RCEdit-500K: Reference Completion for Image-Conditioned Image Editing

Jingxu Zhang[1,2†], Daneul Kim[3], Yueming Pan[2,4], Dong Chen[2], Kai Qiu[2], Yang Liu[2], Yifan Yang[2], Qi Dai[2], Xiaoyan Sun[1*], and Chong Luo[1,2*]

[1] University of Science and Technology of China, Hefei, China
zjxjz@mail.ustc.edu.cn, sunxiaoyan@ustc.edu.cn
[2] Microsoft Research Asia, Beijing, China
{doch, Kai.Qiu, yangliu, yifanyang, Qi.Dai, chong.luo}@microsoft.com
[3] Seoul National University, Seoul, South Korea
carpedkm@snu.ac.kr
[4] Xi'an Jiaotong University, Xi'an, China
pym13474072916@stu.xjtu.edu.cn

Abstract. Image-conditioned image editing (ICIE) guides edits with a reference image to convey visual attributes—such as style, tone, or object identity—that are difficult to specify through text alone. Despite growing practical demand, no large-scale unified ICIE dataset currently exists in the open-source ecosystem; existing efforts cover only a narrow subset of edit types at small scale, typically constructed via costly forward synthesis. We reformulate ICIE data construction as a *reference-completion* problem: high-quality text-conditioned image editing (TCIE) datasets already supply the input image, instruction, and edited target, and can be augmented with aligned reference images through type-specific synthesis and lightweight instruction adaptation. Building on this insight, we propose a scalable pipeline equipped with weak-instruction augmentation and five-dimensional VLM-based post-filtering, and use it to construct **RCEdit-500K**—the first large-scale unified open ICIE dataset comprising 477K quadruplets across six edit categories (add, remove, replace, background, style, alter) with both concrete and abstract reference types. Training on RCEdit-500K consistently improves reference-guided editing: LoRA adaptation on diffusion models yields up to +1.22 average gain, and an autoregressive model without native editing ability acquires competitive ICIE performance, demonstrating that data availability is the primary bottleneck for open-source ICIE.

Keywords: data pipeline · dataset · image-conditioned image editing

1 Introduction

Text-conditioned image editing (TCIE) enables image manipulation through natural-language instructions and has become a widely adopted paradigm. [12,

* Corresponding authors
† Work done during internship at Microsoft Research Asia.

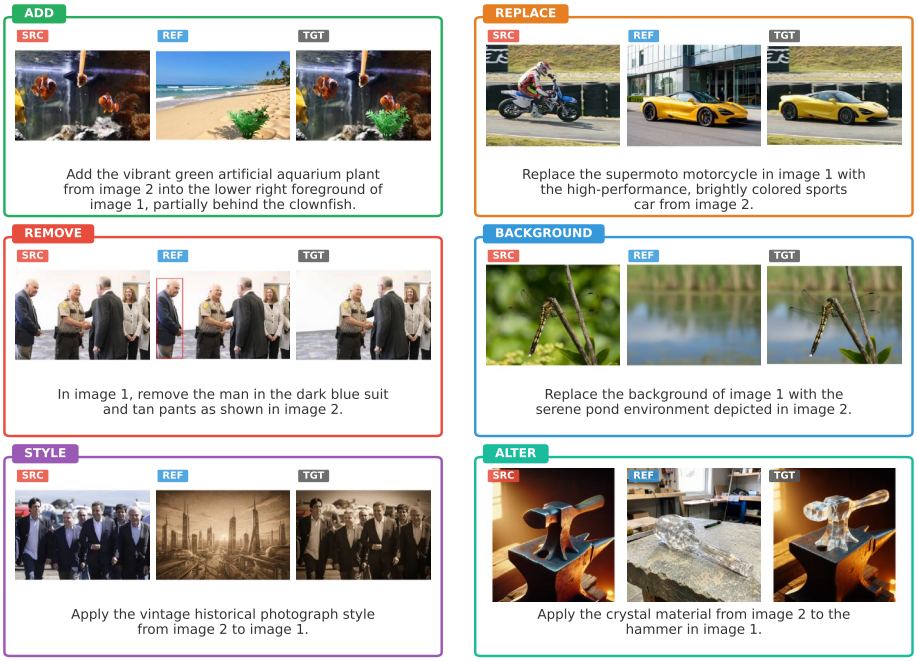

Fig. 1: Examples of six image-conditioned image editing (ICIE) types from RCEdit-500K. Each column shows an input image (left) paired with a reference image (right) and the corresponding edited output. Our dataset enables diverse editing capabilities including object addition, replacement, removal, attribute alteration, style transfer, and background replacement.

13, 33, 40] However, text alone provides a limited channel for specifying many real-world editing objectives. Global visual attributes — such as photographic tone, lighting distributions, material appearance, and stylistic aesthetics — are difficult to describe precisely in language and are often communicated through visual references. This motivates image-conditioned image editing (ICIE), where edits are guided jointly by an input image, a reference image, and a textual instruction.

Despite its practical relevance and demonstrated capabilities in closed-source systems, ICIE remains comparatively underdeveloped in the open-source ecosystem, **with no open-source unified ICIE dataset available**. Existing public datasets [2, 21, 29, 40, 41] and models [12, 13, 33] largely focus on TCIE or on multi-image conditioning for subject-identity preservation [19, 34], while reference-guided editing for distribution-level visual objectives — such as style alignment, tone harmonization, or aesthetic consistency — remains less well supported. As a result, open-source ICIE systems often struggle on these tasks.

To understand this gap, it is important to examine how ICIE training data is typically constructed. A common approach is forward synthesis [37, 40, 41]:

jointly generating input images, reference images, editing instructions, and target outputs using generative models, with at most one image sampled from real data. In practice, this process is expensive and introduces several problems, including model bias, distributional artifacts, and degradation of semantic and aesthetic realism. Crucially, these effects undermine the very qualities — diversity, realism, and data quality — that curated editing datasets aim to preserve.

In contrast, existing high-quality TCIE datasets [21, 29, 40, 41] already provide three of the four elements required for ICIE: an input image, an editing instruction, and a target edited image. The missing component is a compatible reference image. This observation reframes ICIE dataset construction as a reference-completion problem rather than a forward-synthesis task: instead of generating entire editing instances from scratch, one can complete existing TCIE triplets by synthesizing aligned reference images while preserving the original input–target relationship and the edit distribution.

Based on this insight, we introduce a scalable and efficient reference-completion pipeline that converts text-guided image editing datasets into image-conditioned ones. The pipeline performs minimal instruction adaptation and synthesizes reference images that are consistent with the target edit, avoiding joint forward generation and retaining the semantic coverage, aesthetic quality, and realistic edit distributions of the source TCIE corpora. We apply a multi-aspect post-filtering strategy to further enhance the data quality.

Using this pipeline, we construct RCEdit-500K, the first large-scale unified open dataset for image-conditioned image editing containing approximately 500,000 aligned quadruplets: input image, reference image, editing instruction, and target image. The dataset preserves the diversity and realism of curated TCIE sources while enabling structured reference conditioning at scale. **Both the dataset and pipeline will be open-sourced to foster community advancement.**

We validate the effectiveness of RCEdit-500K across multiple model families. With lightweight LoRA adaptation, diffusion-based editors consistently improve on reference-guided editing tasks. Moreover, an autoregressive image-generation model — without explicit editing mechanisms — acquires competitive reference-guided editing behavior when trained on the proposed dataset. These results indicate that training-data construction plays a central role for ICIE and that reference completion provides an effective, scalable foundation for advancing open-source reference-guided image editing. Our contributions are summarized as follows:

- We introduce RCEdit-500K, the first large-scale unified open-source dataset dedicated to image-conditioned image editing, comprising 500K high-quality quadruplets.
- We propose a scalable and efficient data conversion pipeline that transforms existing text-driven image editing data into image-conditioned image editing data, preserving quality strengths from the source corpora while substantially reducing synthetic overhead.

– We provide comprehensive empirical validation across both diffusion-based and autoregressive model families, demonstrating consistent ICIE improvements with RCEdit-500K.

2 Related Work

2.1 Text-Conditioned Image Editing

Text-conditioned image editing (TCIE) has progressed rapidly, with both closed-source models [6, 11, 26] and open-source models [12, 13, 33] achieving strong editing quality. Correspondingly, TCIE datasets have grown in scale and diversity, while maintaining high quality. Early datasets [7, 42] rely on human annotation to ensure quality but remain small in scale. To obtain paired data at larger scale, tool-based automatic pipelines have become the dominant construction paradigm [40, 41, 44], often combined with VLM-based post-filtering to mitigate errors introduced by intermediate models [40]. More recently, pipelines that leverage closed-source editing models for data synthesis [2, 21, 29] have shown that higher-precision data can yield competitive performance with considerably smaller data scale.

2.2 Multi-Image Combination

Multi-image combination, also referred to as multi-image personalization, has attracted growing attention, in part driven by the strong capabilities demonstrated by Nano Banana [6]. A number of recent methods [5, 32, 34, 36] extend single-subject personalization to multi-subject settings. However, these approaches primarily focus on combining concrete subjects and lack support for abstract visual attributes such as style or tone. USO [35] and DreamO [19] address this limitation by generating training pairs that share the same style via style-transfer models. DreamOmni2 [37] further broadens multi-image combination to cover both concrete objects and abstract attributes across a wide distribution. While these methods share the multi-image conditioning paradigm with ICIE, they target image generation rather than editing: the input image is not an explicit anchor whose structure must be preserved.

2.3 Image-Conditioned Image Editing

The need to specify editing operations with precise visual details motivates image-conditioned image editing (ICIE). Several TCIE datasets include a "visual editing" subset as part of their coverage, which shares the same concept with ICIE. ImgEdit [40] produces visual editing data via forward synthesis using an in-context editing model. AnyEdit [41] primarily focuses on structural conditions via ControlNet [43], with only a small portion dedicated to visual reference editing through trained identity and detail extractors. In both cases the ICIE portion remains small ($\sim$10K samples) and restricted to concrete-object addition/replacement or material transfer. DreamOmni2 [37] extends this by training

a concept-extraction model that handles both concrete objects and abstract attributes, enabling forward synthesis of ICIE data across more edit types and concept distributions.

Although several tasks related to ICIE—such as virtual try-on [8,18,28], image structure control [4,27], and interleaved generation [15,34,39]—have individually seen notable progress, each addresses only a specific subtask of ICIE. None provides a unified framework that covers the full spectrum of reference-guided editing types defined in our work. In contrast, this paper focuses on constructing a large-scale dataset that spans all six ICIE edit categories, enabling unified training for reference-guided editing.

3 Dataset

As discussed in Sec. 1, the absence of large-scale unified training data is a key bottleneck for open-source ICIE. In this section, we address this gap by constructing **RCEdit-500K**, shown in Fig. 1. We begin by formalizing the ICIE task and defining a taxonomy of edit types (Sec. 3.1). We then describe our reference-completion pipeline, which converts existing TCIE triplets into ICIE quadruplets by synthesizing aligned reference images and adapting instructions (Sec. 3.2). Finally, we report dataset statistics (Sec. 3.3).

3.1 Task definition

We formulate **image-conditioned image editing** (ICIE) as the task of producing a target image that applies a desired edit to a given **input** (anchor) image under guidance from both a natural-language instruction and a separate **reference** image. Concretely, each ICIE example is represented as a quadruplet

$$\langle \texttt{input\_image}, \texttt{reference\_image}, \texttt{instruction}, \texttt{target\_image} \rangle,$$

where the `input_image` is the anchor whose structural content and unedited regions should be preserved, the `reference_image` supplies visual guidance (a concrete object such as a red handbag or a golden retriever, or an abstract attribute such as style or tone), the `instruction` describes the desired edit in natural language, and the `target_image` is the desired edited output.

ICIE differs from common multi-image generation or multi-reference datasets in that the images do not share equal semantic priority: the input image is explicitly designated as the anchor and must retain the majority of its original structure (including spatial arrangement and any regions outside the edit), while the reference image functions purely as a conditioning signal.

Following prior work [40], we categorize ICIE edits into six types: **add**, **remove**, **replace**, **background**, **style**, and **alter**. We adopt this taxonomy because it cleanly separates semantic edits (add/replace/remove) from distributional or appearance-level edits (background/style/alter), which is useful when designing evaluation protocols and conditioning mechanisms. We intentionally

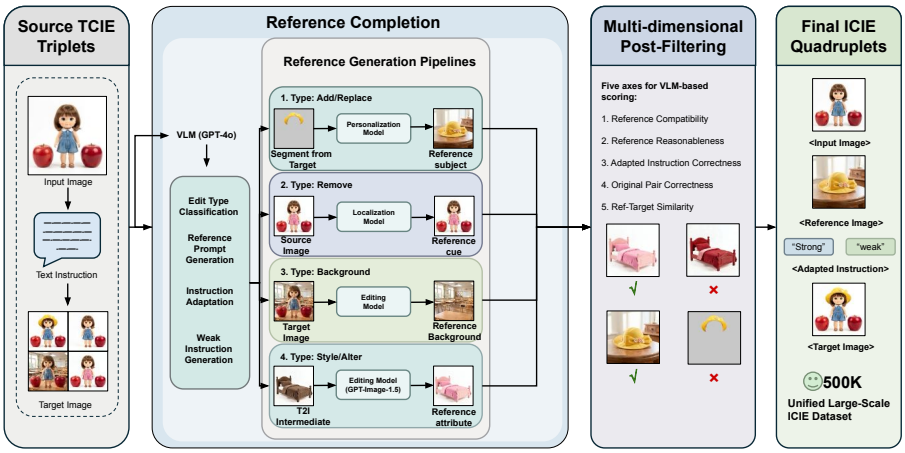

Fig. 2: Data curation pipeline to construct RCEdit-500K.

exclude a separate **motion** category from our ICIE definition since it is hard to describe with single reference (see supplementary for taxonomy details).

The **remove** category merits special attention. In scenes containing many instances of the same object class, purely textual instructions (e.g., "remove the person") are often ambiguous. For ICIE we therefore recommend supplementing the instruction with an explicit spatial cue derived from the input image. Acceptable forms of such cues include a binary mask, a tight bounding box, or a cropped exemplar of the object to be removed; each of these options disambiguates which instance should be edited while remaining compatible with image-conditioned modeling. For example, a removal request may be specified as "remove the object indicated by the bounding box" or by providing a cropped patch of the object to be excised.

3.2 Data Curation Pipeline

Source Data Preparation. To the best of our knowledge, no publicly available unified dataset currently exists for ICIE, making it essential to construct one. A key observation is that existing high-quality TCIE datasets already contain three of the four elements required for ICIE: ⟨`input_image`, `instruction`, `target_image`⟩; only the reference image is missing. As discussed in Sec. 3.1, every ICIE edit type corresponds to a valid TCIE edit, meaning that ICIE dataset construction can be naturally cast as a **reference-completion** problem rather than a full forward-synthesis task.

We select source data from two large-scale, high-quality TCIE corpora: the complete single-turn split of Pico-Banana-400K [21] and a 500K-sample subset of GPT-IMAGE-EDIT-1.5M [29]. Together, these two sources provide broad

coverage of semantic and appearance-level edits while preserving high-quality edit precision.

Reference Completion. The data construction pipeline is shown in Fig. 2. For each source triplet, we first employ a vision-language model (VLM; specifically GPT-4o [11]) to (i) classify the edit into one of the six defined types, (ii) generate several prompts for reference-image synthesis, and (iii) rewrite the original text instruction so that it explicitly references the supplementary image. Triplets whose edits fall outside the defined taxonomy (*e.g.* motion editing) are discarded at this stage.

In addition to the adapted instruction, we ask the VLM to produce a **weak** textual instruction in which the reference-specific description is deliberately abstracted. For example, "*Add the cute orange cat from image 2 to image 1*" is weakened to "*Add the animal from image 2 to image 1*," removing fine-grained appearance details that could be resolved from text alone. By mixing weak instructions into training, we encourage the model to attend more heavily to the reference image as a visual conditioning signal, rather than relying solely on its pre-existing text-conditioned editing capability.

We then design four dedicated pipelines to synthesize reference images according to edit type:

1. **Add / Replace.** The target subject is extracted from the target image using a segmentation model [16, 23, 24]. A reference image is then generated either by re-synthesizing the subject via an image personalization model [12] or by compositing the segmented subject onto a novel background.
2. **Remove.** The subject to be removed is localized in the input image with the same segmentation model [16, 23, 24]. A reference cue is produced by overlaying a tight bounding box on the input image or by cropping the target region, providing an unambiguous spatial indicator of which instance to remove.
3. **Background.** A TCIE model is applied to inpaint all foreground objects while retaining the desired background, yielding a clean background reference that captures the target scene layout and appearance.
4. **Style / Alter.** An intermediate image is first generated by a text-to-image (T2I) model from a descriptive prompt. This intermediate is then transformed to exhibit the required abstract attribute (*e.g.* artistic style, color palette, texture) using GPT-Image-1.5.

We use Grounded-SAM-2 [16, 23, 24] for localization and segmentation, and Flux-Klein-9B [12] for personalization, T2I generation, and TCIE. For style and alter edits, open-source models do not yet achieve sufficient quality for abstract attribute transfer; we therefore employ GPT-Image-1.5 for these categories to ensure data fidelity. **The choices of our toolkit enables data harvesting with low-capacity machines like V100.** Full VLM prompts and synthesis details are in the supplementary.

Post-Filtering. After reference completion, we apply a multi-dimensional quality filter to remove low-quality samples. Prior image edit data-construction pipelines [40, 41] typically rely on CLIP [22] similarity or a holistic VLM score,

Table 1: Post-filter validation. Cohen's κ: agreement beyond chance (1=perfect, 0=random).

Setting	**Comparison**	Agreement↑	κ↑
Cross-VLM (N=600)	GPT-4o vs. Claude 4.6	72.9%	0.46
	GPT-4o vs. Gemini 3.1	78.6%	0.57
Human (10 ann., N=60)	Mean human–human	73.1%	0.38
	Mean GPT-4o–human	72.2%	0.44
GPT-4o Precision / Recall / F1		96.7 / 76.3 / 85.3%	

which conflates multiple quality axes and leads to incomplete quality assessment. Similarly, existing benchmark evaluation protocols [20] focus primarily on the fidelity of the target image, without verifying consistency across all four elements of the quadruplet.

To address these limitations, we propose a **five-dimensional VLM-based filtering** strategy that independently scores each candidate quadruplet on: (i) **reference compatibility for original TCIE pair**, (ii) **reference image reasonableness**, (iii) **adapted instruction correctness**, (iv) **original pair correctness**, and (v) **reference–target similarity**. Detailed scoring prompts and threshold selections are provided in the supplementary material.

Each dimension captures a distinct failure mode in reference-conditioned editing. A sample is retained only if **every** dimension exceeds a predefined threshold; a low score on any single axis suffices for rejection. This strict, axis-wise filtering ensures that the final dataset maintains high consistency across all components of the quadruplet.

To validate the reliability of VLM-based filtering, we conduct cross-VLM and human-agreement studies (Tab. 1). Re-running the 5-axis filter on 600 stratified samples with Claude Opus 4.6 and Gemini 3.1 Pro yields cross-VLM Cohen's κ that meets or exceeds human inter-annotator κ. A 10-annotator human study on 60 samples confirms that GPT-4o–human agreement exceeds mean human–human agreement, with high precision and moderate recall, confirming a conservative filter that rarely retains flawed samples.

Pipeline Comparison. Fig. 3 compares our reference-completion pipeline with two alternative paradigms: forward synthesis and random combination. Forward synthesis constructs all four quadruplet components sequentially, requiring 2–3 image-model calls per sample; this compounds errors and limits scalability. Random combination pairs each input with a randomly sampled reference, achieving high volume scalability but suffering from low semantic compatibility between input and reference, leading to high failure rates.

Our reference-completion pipeline inherits the triplets from curated TCIE corpora, which makes it efficient and ensure high data success rate and reasonableness. The dataset scale can be enriched by incorporating more TCIE data. Moreover, while both forward synthesis and random combination are constrained by pre-defined keyword pools or image pools and can only benefit from basic

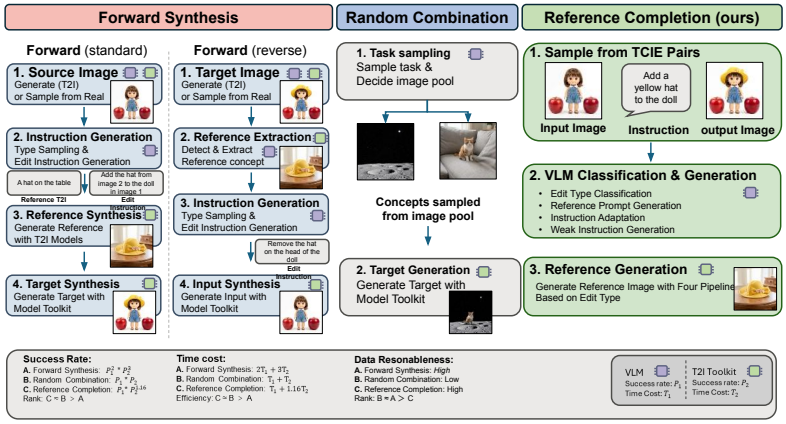

Fig. 3: Comparison of three ICIE data construction paradigms. Forward synthesis generates all quadruplet components from a single starting image; random combination samples references from an image pool; our reference completion inherits curated TCIE triplets and synthesizes only the missing reference image.

Table 2: Comparison of image-conditioned image-editing datasets. RCEdit-500K is larger in scale and provides a unified taxonomy covering all edit types.

	Data Size	Resolution	Concrete	Abstract	Type					
					Add	Remove	Replace	Background	Style	Alter
ImgEdit [40]	60K	1024	✓	×	✓	×	✓	×	×	×
AnyEdit [41]	40K	512	✓	✓	✓	×	✓	×	×	✓
RCEdit-500K (Ours)	477K	1024	✓	✓	✓	✓	✓	✓	✓	✓

model improvements, our pipeline can further enhance data quality by adopting stronger model toolkits and leveraging better TCIE datasets as they become available.

As shown in Tab. 2, although some TCIE datasets [40, 41] include a "visual editing" subset that overlaps with ICIE, their coverage is limited to concrete-object edits (with only a small material-transfer subset in AnyEdit [41]) and remains at a small scale. In contrast, RCEdit-500K provides 477K samples at 1024 resolution, covering all six edit categories with both concrete and abstract reference types.

3.3 Data Statistics

Fig. 4 and Fig. 5 summarize the statistics of RCEdit-500K. We use LAION-Aesthetics-Predictor-V1 [25] to estimate aesthetic scores, and employ patch-wise SSIM [30] to quantify edit-region size: for each image pair, we compute per-patch SSIM between input and target and apply k-means [17] clustering to separate high- and low-similarity patches, with the low-similarity cluster indicating the edited region.

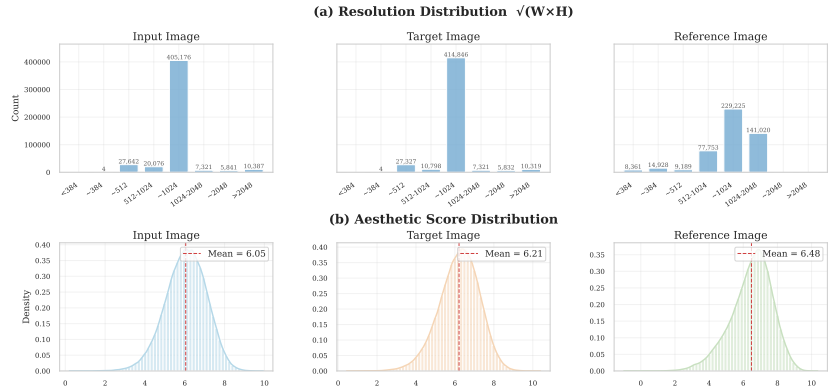

Fig. 4: Data statistics of RCEdit-500K: (a) resolution distribution and (b) aesthetic score distribution.

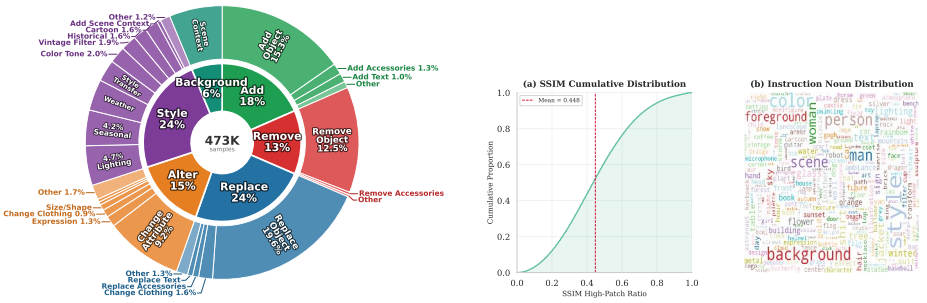

Fig. 5: Left: Edit-type diversity of RCEdit-500K—6 coarse categories and 35 fine-grained types classified by GPT-4o (validated accuracy 92.7%); no coarse type exceeds 24%. Right: Patch-wise SSIM distribution and word cloud; the SSIM patch ratio measures the fraction of patches with high similarity between input and target images.

As shown in Fig. 4, the resolution distributions are well-balanced and consistent with those of recent TCIE datasets [2, 40]. The majority of images achieve aesthetic scores above 4, with a mean of approximately 6, indicating high visual quality. Fig. 5 (right) shows that the edit-region ratio is neither too small nor too large for most samples, providing a healthy distribution for model training. The word cloud further illustrates the lexical diversity of instructions in RCEdit-500K.

Among the six edit categories, add and replace together account for the largest share, reflecting the dominance of subject-centric edits in the source TCIE corpora. Style and alter edits, while smaller in absolute count, still comprise a substantial portion, ensuring adequate coverage of abstract-attribute conditioning—a capability largely absent from prior ICIE datasets (Tab. 2). As shown in Fig. 5 (left), 35 fine-grained sub-types are identified by GPT-4o classification, with no single coarse type exceeding 24% of the total, confirming balanced coverage

across edit semantics. The post-filtering step discards approximately 12.5% of candidates, with the highest rejection rates observed for style and alter types, where synthesized references are most prone to semantic mismatch. This confirms that our multi-dimensional filtering is especially valuable for the more challenging abstract-attribute edits.

4 Experiments

We validate the effectiveness of RCEdit-500K by fine-tuning multiple models and evaluating their ICIE performance. Sec. 4.1 describes the training configurations, Sec. 4.2 introduces the evaluation benchmark, Sec. 4.3 presents quantitative and qualitative results, and Sec. 4.4 ablates the contributions of post-filtering and weak-instruction.

4.1 Experimental Settings

We fine-tune two diffusion models with LoRA [9]: Flux-Kontext [13] and Qwen-Image-Edit-2509 [33]. These two models are chosen to examine whether our dataset benefits models at different capability levels—Flux-Kontext has weak baseline ICIE performance, whereas Qwen-Image-Edit-2509 already exhibits reasonable editing ability. Both LoRA rank and alpha are set to 256. For the autoregressive family, we select Janus-Pro [3] and perform full fine-tuning. During inference on ICIE tasks, we concatenate image tokens from both the generation encoder and the understanding encoder to preserve editing fidelity. Further implementation details are provided in the supplementary material.

4.2 Evaluation Setups

Evaluation protocols for ICIE remain under-explored [14]. Traditional pairwise similarity metrics (*e.g.* CLIP [22] similarity, SSIM [30]) are inadequate because they do not account for multi-image conditioning. VLM-based evaluation is therefore more appropriate [40]. We adopt MultiBanana [20] as our benchmark, which uses a VLM to assess multi-reference text-to-image generation along five axes: Instruction Alignment (IA), Reference Consistency (RC), Background-Subject Match (BSM), Physical Realism (PR), and Visual Quality (VQ). We evaluate on its two-reference subset, which directly corresponds to the ICIE setting. The "makeup" category is excluded from our evaluation for privacy considerations. All VLM evaluations are conducted with GPT-4o.

4.3 Results

Quantitative Results. Tab. 3 reports the MultiBanana two-reference evaluation results. The closed-source GPT-Image-1.5 attains the highest overall score by a clear margin, underscoring the persistent gap between proprietary and open-source ICIE systems.

Table 3: Quantitative comparison on the MultiBanana two-reference subset. Please refer to Sec. 4.2 for metric full-names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Closed-source models						
Nano Banana [6]	4.59	4.00	5.13	5.72	5.44	4.67
GPT-Image-1.5	5.87	4.94	5.40	5.89	5.85	5.51
Open-source models						
Flux-Kontext [13]	2.94	2.80	2.39	2.94	2.81	2.82
Omnigen2 [34]	3.62	3.28	4.50	5.17	5.05	3.93
Qwen-Image-Edit-2509 [33]	3.60	4.06	4.83	5.61	5.39	4.31
Dreamomni2 [37]	4.45	3.61	5.07	5.83	5.80	4.54
Qwen-Image-Edit-2511 [33]	4.15	4.18	4.95	5.68	5.59	4.58
Flux-Klein-4B [12]	4.69	4.01	5.05	5.62	5.57	4.71
Flux-Klein-9B [12]	4.78	4.09	5.01	5.61	5.50	4.75
Janus-Pro	1.02	1.01	1.02	1.04	1.11	1.03
+ RCEdit-500K (Ours)	4.65	3.81	4.52	5.04	4.91	4.43
Flux-Kontext	2.94	2.80	2.39	2.94	2.81	2.82
+ RCEdit-500K (Ours)	3.52	3.42	4.76	5.50	5.27	4.04
Qwen-Image-Edit-2509	3.60	4.06	4.83	5.61	5.39	4.31
+ RCEdit-500K (Ours)	4.08	4.03	5.11	5.86	5.72	4.56

Fine-tuning on RCEdit-500K yields consistent gains across both diffusion models. For Flux-Kontext, LoRA adaptation raises the average by +1.22, demonstrating that RCEdit-500K can transform a model with weak ICIE capability into a competitive one through parameter-efficient finetuning. For Qwen-Image-Edit-2509, which already possesses reasonable editing ability, LoRA fine-tuning brings moderate but consistent improvement across all axes. The gain on Instruction Alignment further confirms that RCEdit-500K strengthens the model's capacity to follow reference-conditioned instructions. Notably, this lightweight adaptation achieves performance comparable to the newer Qwen-Image-Edit-2511, which was full-fine-tuned on large-scale data.

For the autoregressive model, Janus-Pro-ICIE, fully fine-tuned on our dataset, surpasses both Qwen-Image-Edit-2509 and Omnigen2. Its Instruction Alignment ranks third overall, revealing strong instruction-following potential in autoregressive architectures. This demonstrates that autoregressive models, despite not being originally designed for image editing, can acquire competitive reference-guided editing capabilities when trained on high-quality ICIE data—reinforcing that data availability is the primary bottleneck for ICIE.

Qualitative Results. Fig. 6 provides qualitative comparisons between each base model and its RCEdit-500K fine-tuned counterpart, spanning both concrete-object edits (add, replace) and abstract-attribute edits (style, alter). For Qwen-Image-Edit-2509, the original model tends to copy-paste the reference image, largely ignoring the instruction; after fine-tuning, instruction alignment improves substantially, consistent with the IA gain in Tab. 3. Flux-Kontext originally

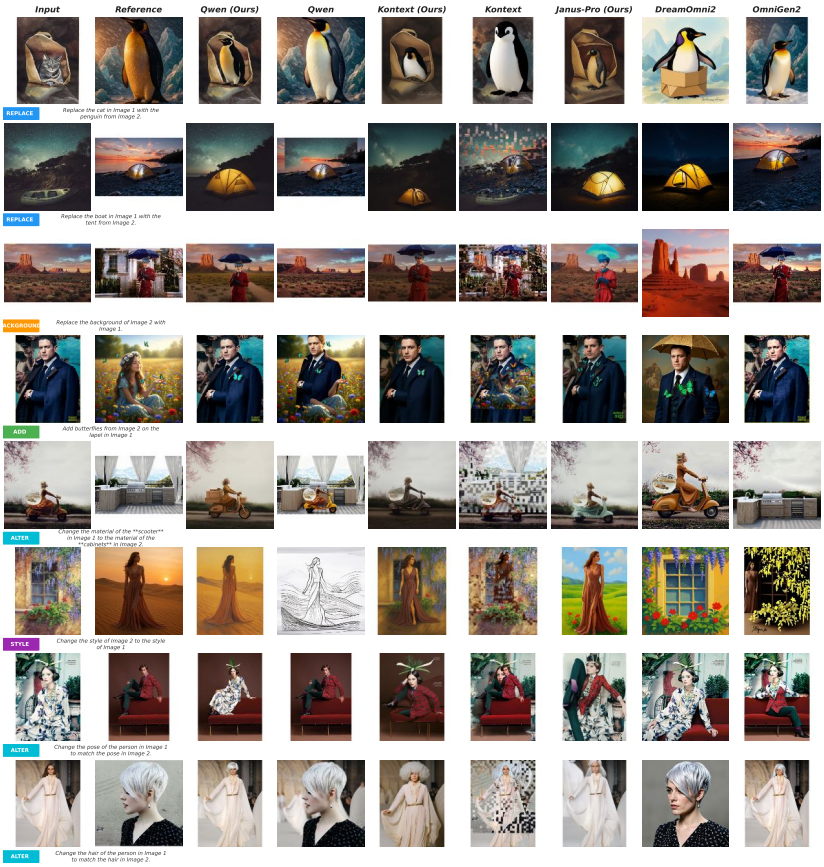

Fig. 6: Qualitative comparison of different models. Models labeled "Ours" are trained with RCEdit-500K, demonstrating consistent performance improvements across various tasks. "Qwen" refers to Qwen-Image-Edit-2509, and "Kontext" refers to Flux-Kontext.

suffers from visual blending between input and reference, producing incoherent outputs; this artifact is strongly reduced after fine-tuning. Janus-Pro trained on RCEdit-500K produces successful edits despite occasional artifacts from its limited base capability.

4.4 Ablation Study

We ablate two key designs in our data curation pipeline—post-filtering and weak-instruction augmentation (Tab. 4)—and compare against existing ICIE datasets at both matched and full scale to validate data quality and scalability (Tab. 5). All experiments use Flux-Kontext LoRA as the base configuration.

Data Post-Filtering. Removing the five-dimensional VLM-based post-filtering (Sec. 3.2) causes the largest overall degradation. The most pronounced declines

Table 4: Ablation study on post-filtering and weak-instruction augmentation evaluated on MultiBanana two-reference subset. All variants use Flux-Kontext with LoRA fine-tuning on RCEdit-500K.

Variant	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
w/o post-filtering	3.60	3.30	3.62	4.20	4.04	3.62
w/o weak instruction	**3.67**	3.35	3.84	4.50	4.27	3.74
full	3.52	**3.42**	**4.76**	**5.50**	**5.27**	**4.04**

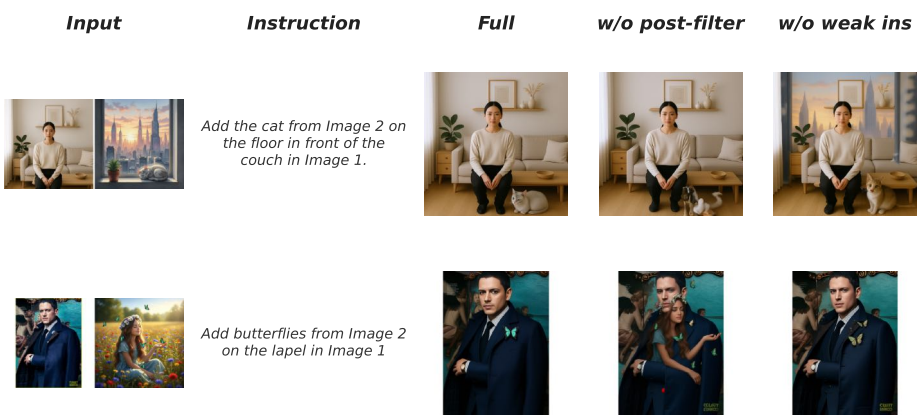

Fig. 7: Qualitative comparison of ablation study.

appear in BSM, PR, and VQ, indicating that unfiltered quadruplets introduce noise that severely impairs the model's ability to produce physically plausible and visually coherent outputs. In contrast, IA and RC are less affected, suggesting that these axes are more robust to data noise. As illustrated in Fig. 7, training without post-filtering leads to noticeably more artifacts in the generated images, validating that our multi-dimensional filtering strategy is critical for maintaining high data quality.

Weak Instruction. Removing weak-instruction augmentation also degrades overall performance, with the drops again concentrated on BSM, PR, and VQ. Without weak instructions, the model can partially resolve the edit from text alone, reducing its reliance on the reference image as a visual conditioning signal. This explains the slight IA increase alongside the RC decrease: the model follows the textual instruction but attends less to the reference's visual content. Qualitatively, Fig. 7 shows that although the model correctly performs the requested edit (*e.g.* adding a cat or butterfly), the generated subject differs noticeably from the one depicted in the reference image, confirming that the model falls back on its pre-existing TCIE capability. This validates our design motivation in Sec. 3.2:

Table 5: Ablation on training data. All models use Flux-Kontext LoRA. Baseline is without LoRA.

Training Data	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Baseline	2.94	2.80	2.39	2.94	2.81	2.82
ImgEdit	3.05	2.83	2.58	3.10	2.92	2.92
AnyEdit (w/o structure)	3.02	2.65	2.10	2.57	2.44	2.68
AnyEdit	2.17	1.87	4.55	5.15	4.93	2.97
RCEdit-60K (Ours)	**3.62**	3.28	3.78	4.45	4.16	3.68
RCEdit-500K (Ours)	3.52	**3.42**	**4.76**	**5.50**	**5.27**	**4.04**

deliberately abstracting reference-specific details in the instruction forces the model to ground its edits in the reference image, yielding stronger visual fidelity.

Comparison with Existing ICIE Datasets. We compare training on existing ICIE subsets—ImgEdit [40] and AnyEdit [41]—against RCEdit variants, including **RCEdit-60K** for scale-controlled comparison.

As shown in Tab. 5, existing ICIE subsets yield only marginal gains over the baseline. Notably, AnyEdit severely degrades IA and RC despite boosting BSM, PR, and VQ, suggesting the model defaults to unconditional generation rather than genuine reference-guided editing. The contrast with RCEdit is striking: at matched scale, RCEdit-60K outperforms both by a large margin while improving all five metrics, demonstrating that only high-quality ICIE data yields meaningful gains. Scaling to RCEdit-500K yields further improvement, confirming that our reference-completion pipeline scales favorably with the size of the underlying TCIE corpus. Per-type results are in the supplementary.

5 Conclusion

We introduce **RCEdit-500K**, the first large-scale unified open-source dataset for image-conditioned image editing (ICIE). By reframing ICIE data construction as a reference-completion problem, our pipeline converts existing TCIE triplets into ICIE quadruplets by synthesizing only the missing reference image and adapting instructions accordingly. Combined with five-dimensional VLM-based post-filtering and weak-instruction augmentation, this yields 477K high-quality quadruplets spanning six edit categories. Experiments across diffusion-based and autoregressive models demonstrate consistent improvements, with even an autoregressive model acquiring competitive ICIE capabilities—reinforcing that **data availability** is the primary bottleneck for ICIE. The dataset is publicly available at `https://huggingface.co/datasets/carpedkm/RCEdit-500K`.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62472399 and the Open Fund of APKL of BIIP, IAI, Hefei Comprehensive National Science Center under Grant 24YGXT003.

References

1. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
2. Chen, J., Cai, Z., Chen, P., Chen, S., Ji, K., Wang, X., Yang, Y., Wang, B.: Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. arXiv preprint arXiv:2506.18095 (2025)
3. Chen, X., Wu, Z., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C.: Janus-pro: Unified multimodal understanding and generation with data and model scaling. arXiv preprint arXiv:2501.17811 (2025)
4. Chen, Y., Ma, Z., Jia, G., Jiang, C., Li, J., Zhou, B.: Context-aware autoregressive models for multi-conditional image generation. arXiv preprint arXiv:2505.12274 (2025)
5. Cheng, Y., Wu, W., Wu, S., Huang, M., Ding, F., He, Q.: Umo: Scaling multi-identity consistency for image customization via matching reward. arXiv preprint arXiv:2509.06818 (2025)
6. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
7. Ge, Y., Zhao, S., Li, C., Ge, Y., Shan, Y.: Seed-data-edit technical report: A hybrid dataset for instructional image editing. arXiv preprint arXiv:2405.04007 (2024)
8. He, Z., Ning, Y., Qin, Y., Wang, G., Yang, S., Lin, L., Li, G.: Vton 360: High-fidelity virtual try-on from any viewing direction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26388–26398 (2025)
9. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022)
10. Hui, M., Yang, S., Zhao, B., Shi, Y., Wang, H., Wang, P., Zhou, Y., Xie, C.: Hq-edit: A high-quality dataset for instruction-based image editing. arXiv preprint arXiv:2404.09990 (2024)
11. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
12. Labs, B.F.: FLUX.2: Frontier Visual Intelligence (2025)
13. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025)
14. Li, R., Qu, L., Zhang, J., Gui, D., Xu, M., Zhang, X., Hu, H., Wang, W., Wang, J.: Genarena: How can we achieve human-aligned evaluation for visual generation tasks? arXiv preprint arXiv:2602.06013 (2026)
15. Li, S., Gu, J., Liu, K., Lin, Z., Wei, Z., Grover, A., Kuen, J.: Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation. arXiv preprint arXiv:2509.19244 (2025)
16. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)

17. McQueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proc. of 5th Berkeley Symposium on Math. Stat. and Prob. pp. 281–297 (1967)
18. Mei, S., Ni, B.: Physdiff-vton: Cross-domain physics modeling and trajectory optimization for virtual try-on. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
19. Mou, C., Wu, Y., Wu, W., Guo, Z., Zhang, P., Cheng, Y., Luo, Y., Ding, F., Zhang, S., Li, X., et al.: Dreamo: A unified framework for image customization. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–12 (2025)
20. Oshima, Y., Miyake, D., Matsutani, K., Iwasawa, Y., Suzuki, M., Matsuo, Y., Furuta, H.: Multibanana: A challenging benchmark for multi-reference text-to-image generation. arXiv preprint arXiv:2511.22989 (2025)
21. Qian, Y., Bocek-Rivele, E., Song, L., Tong, J., Yang, Y., Lu, J., Hu, W., Gan, Z.: Pico-banana-400k: A large-scale dataset for text-guided image editing. arXiv preprint arXiv:2510.19808 (2025)
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
23. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos (2024)
24. Ren, T., Jiang, Q., Liu, S., Zeng, Z., Liu, W., Gao, H., Huang, H., Ma, Z., Jiang, X., Chen, Y., Xiong, Y., Zhang, H., Li, F., Tang, P., Yu, K., Zhang, L.: Grounding dino 1.5: Advance the "edge" of open-set object detection (2024)
25. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in neural information processing systems **35**, 25278–25294 (2022)
26. Seedream, T., Chen, Y., Gao, Y., Gong, L., Guo, M., Guo, Q., Guo, Z., Hou, X., Huang, W., Huang, Y., et al.: Seedream 4.0: Toward next-generation multimodal image generation. arXiv preprint arXiv:2509.20427 (2025)
27. Sun, H., He, Q., Peng, J., Tang, P., Zhang, J., Zhu, J., Hu, X., Yan, S.: Tokenar: Multiple subject generation via autoregressive token-level enhancement. arXiv preprint arXiv:2510.16332 (2025)
28. Wan, S., Chen, J., Cai, Q., Pan, Y., Yao, T., Mei, T.: Vton-vllm: Aligning virtual try-on models with human preferences. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
29. Wang, Y., Yang, S., Zhao, B., Zhang, L., Liu, Q., Zhou, Y., Xie, C.: Gpt-image-edit-1.5m: A million-scale, gpt-generated image dataset (2025)
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
31. Wei, C., Xiong, Z., Ren, W., Du, X., Zhang, G., Chen, W.: Omniedit: Building image editing generalist models through specialist supervision. In: The Thirteenth International Conference on Learning Representations (2024)
32. Wei, Y., Zhang, S., Yuan, H., Gong, B., Tang, L., Wang, X., Qiu, H., Li, H., Tan, S., Zhang, Y., et al.: Dreamrelation: Relation-centric video customization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12381–12393 (2025)

33. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025)
34. Wu, C., Zheng, P., Yan, R., Xiao, S., Luo, X., Wang, Y., Li, W., Jiang, X., Liu, Y., Zhou, J., et al.: Omnigen2: Exploration to advanced multimodal generation. arXiv preprint arXiv:2506.18871 (2025)
35. Wu, S., Huang, M., Cheng, Y., Wu, W., Tian, J., Luo, Y., Ding, F., He, Q.: Uso: Unified style and subject-driven generation via disentangled and reward learning. arXiv preprint arXiv:2508.18966 (2025)
36. Wu, S., Huang, M., Wu, W., Cheng, Y., Ding, F., He, Q.: Less-to-more generalization: Unlocking more controllability by in-context generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18682–18692 (2025)
37. Xia, B., Peng, B., Zhang, Y., Huang, J., Liu, J., Li, J., Tan, H., Wu, S., Wang, C., Wang, Y., et al.: Dreamomni2: Multimodal instruction-based editing and generation. arXiv preprint arXiv:2510.06679 (2025)
38. Ye, K., Huang, Z., Fu, C., Liu, Q., Cai, J., Lv, Z., Li, C., Lyu, J., Zhao, Z., Zhang, S.: Unicedit-10m: A dataset and benchmark breaking the scale-quality barrier via unified verification for reasoning-enriched edits. arXiv preprint arXiv:2512.02790 (2025)
39. Ye, M., Liu, J., Song, Y.: Loom: Diffusion-transformer for interleaved generation. arXiv preprint arXiv:2512.18254 (2025)
40. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2025)
41. Yu, Q., Chow, W., Yue, Z., Pan, K., Wu, Y., Wan, X., Li, J., Tang, S., Zhang, H., Zhuang, Y.: Anyedit: Mastering unified high-quality image editing for any idea. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26125–26135 (2025)
42. Zhang, K., Mo, L., Chen, W., Sun, H., Su, Y.: Magicbrush: A manually annotated dataset for instruction-guided image editing. Advances in Neural Information Processing Systems **36**, 31428–31449 (2023)
43. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023)
44. Zhao, H., Ma, X.S., Chen, L., Si, S., Wu, R., An, K., Yu, P., Zhang, M., Li, Q., Chang, B.: Ultraedit: Instruction-based fine-grained image editing at scale. Advances in Neural Information Processing Systems **37**, 3058–3093 (2024)

A Timeline of TCIE Datasets

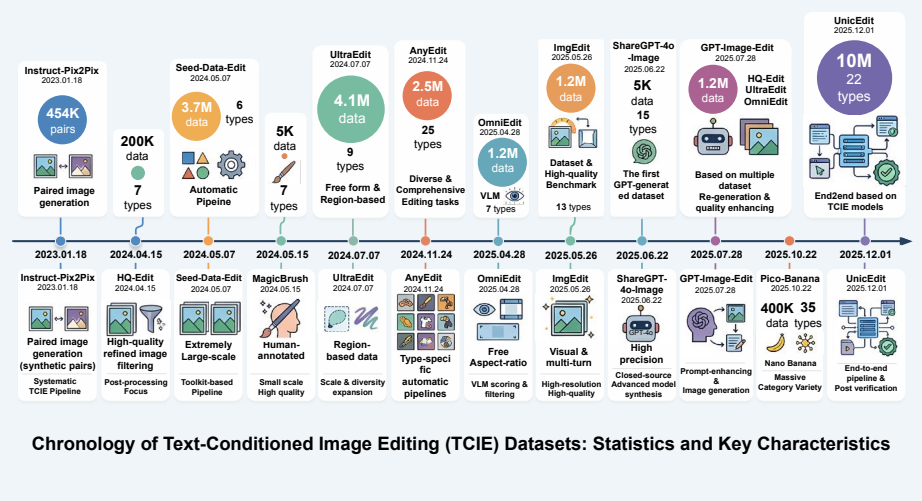

Fig. 8: Timeline of text-conditioned image editing (TCIE) datasets, showing the evolution of data scale, number of edit types, and key dataset characteristics. TCIE corpora have steadily grown in scale, type coverage, precision, resolution, and quality.

Fig. 8 illustrates the development timeline of TCIE datasets. Early works such as Instruct-Pix2Pix [1] and HQ-Edit [10] relied on synthetic paired image generation with limited editing types and moderate scale, while concurrent efforts [7, 42] focused on small-scale human annotation to improve quality.

Subsequent datasets introduced automated pipelines and editing toolkits (*e.g.* Seed-Data-Edit [7] and UltraEdit [44]), significantly expanding the data scale. More recent efforts further improve diversity and quality by incorporating richer editing taxonomies, region-based editing, and broader task coverage (*e.g.* AnyEdit [41] and ImgEdit [40]), extending the toolkit-based paradigm with type-specific pipelines for each edit category. In parallel, works such as ShareGPT-4o-Image [2] demonstrate that data constructed with advanced closed-source models [11] achieves high precision, yielding competitive training results even at smaller scale. Large language models and multimodal models are increasingly used to synthesize and refine editing data, enabling high-quality large-scale datasets such as GPT-Image-Edit [29] and Pico-Banana [21]. UnicEdit [38] further pushes the frontier to the ten-million scale.

Overall, TCIE datasets exhibit a clear trend toward larger scale, richer editing types, higher visual quality, and more automated generation pipelines. This growing ecosystem of curated TCIE corpora provides a strong foundation for our reference-completion approach to ICIE dataset construction.

B Details of Data Pipeline

B.1 Details of Reference Completion

As described in Sec. 3.2, the reference-completion pipeline operates in two stages: (1) a single GPT-4o call that performs edit-type classification, reference-prompt generation, and instruction adaptation, and (2) type-specific reference-image synthesis using the generated prompts. We describe each stage in detail below, with the overall flow illustrated in Fig. 2.

Stage 1: VLM Analysis (Single GPT-4o Call). The prompt for this stage is shown in Appendix G. Given a source TCIE triplet ⟨input image, target image, text instruction⟩, we make a single call to GPT-4o that jointly performs all of the following:

1. **Edit-type classification.** The VLM classifies the edit into one of {add, replace, remove, background, style, alter, others} based on the input–target pair and the text instruction. Entries classified as "others" are discarded.
2. **Assertion check.** The VLM verifies that the edit is suitable for single-reference ICIE. Edits involving multiple objects to be added, replaced, or removed simultaneously are flagged as invalid (assertion = false) and discarded, since they would require multiple reference images.
3. **Reference-prompt generation.** Depending on the classified type, the VLM generates the prompts needed for reference-image synthesis:
   - **Add / Replace:** a ***segmentation keyword*** to extract the target subject, a ***personalization prompt*** to re-synthesize the subject in a novel context, and an ***edit prompt*** to change the subject's background.
   - **Remove:** a ***segmentation keyword*** identifying the object to be removed from the input image.
   - **Background:** an ***edit prompt*** to remove all foreground objects from the target image, yielding a clean background.
   - **Style / Alter:** a ***generation prompt*** to create an intermediate image with different identity and appearance from the target, and an ***edit prompt*** to transfer the target's style or attribute onto this intermediate.
4. **Instruction adaptation.** The VLM rewrites the original text instruction to explicitly reference the supplementary image (*e.g.* "Add a zebra beside the man" → "Add the zebra from image 2 beside the man in image 1").
5. **Weak instruction generation.** The VLM also produces a weakened version of the adapted instruction by replacing specific reference descriptions with generic terms (*e.g.* "zebra" → "animal", "oil painting style" → "style"), as described in Sec. 3.2.

Stage 2: Type-Specific Reference Synthesis. Using the prompts generated in Stage 1, we synthesize reference images through four dedicated pipelines. All images are processed to around 1024×1024 pixels while retaining original aspect ratio.

1. **Add / Replace.** Grounded-SAM-2 [16, 23, 24] extracts the target subject from the target image using the ***segmentation keyword***. The segmented subject (placed on a gray background) is then processed by Flux-Klein-9B [12] in one of two modes: (a) *personalization mode*, which re-synthesizes the subject in a novel scene or pose while preserving identity, or (b) *background-edit mode*, which replaces the gray background with a contextually different scene. Both modes use the ***prompts*** generated by GPT-4o.
2. **Remove.** Grounded-SAM-2 localizes the object to be removed in the *input* image. The reference cue is produced by either overlaying a tight bounding box on the input image or cropping the localized region, providing an unambiguous spatial indicator.
3. **Background.** Flux-Klein-9B operates in TCIE mode on the *target* image, using the ***edit prompt*** to inpaint all foreground objects while preserving the background. The result serves as a clean background reference.
4. **Style / Alter.** This type requires an additional closed-source model call and proceeds in two steps:
   (a) *Intermediate generation:* Flux-Klein-9B generates an intermediate image from the ***generation prompt***. This image has a different identity, appearance, and background from the target.
   (b) *Attribute transfer:* GPT-Image-1.5 receives the intermediate image and the target image, and transfers the target's style or attribute onto the intermediate using the ***edit prompt***. The result becomes the reference image.

   GPT-Image-1.5 is used because open-source models do not yet achieve sufficient quality for abstract attribute transfer. This step is the only part of the pipeline that requires a closed-source model.

B.2 Details of Post Filtering

As described in Sec. 3.2, we apply a five-dimensional VLM-based filtering strategy to each candidate quadruplet. Each dimension is scored independently on a 0–100 scale by GPT-4o. Below we describe the five evaluation axes in detail:

1. **Reference Compatibility (A).** Evaluates whether the original TCIE editing task naturally supports the introduction of a reference image. Tasks that inherently require multiple references (*e.g.* camera control, position change) or gain no benefit from visual conditioning receive low scores. This dimension guards against forcing reference conditioning onto edits where it is fundamentally unsuitable.
2. **Reference Image Reasonableness (B).** Assesses whether the synthesized reference image provides relevant and transferable visual information for the intended edit, without being misleading or irrelevant. The visual quality of the reference is also considered. A high score indicates that the reference conveys the correct concept (object identity, style, or attribute) needed to guide the edit.

3. **Instruction Correctness (C).** Checks whether the adapted instruction accurately describes how the reference image should be used, without requiring information that the reference does not contain. This ensures that the instruction-reference pair forms a consistent and learnable conditioning signal.
4. **Original Pair Correctness (D).** Verifies whether the original target image correctly satisfies the original TCIE instruction, independent of the reference image. Since our pipeline inherits input–target pairs from source TCIE corpora, this dimension filters out cases where the source data itself is flawed.
5. **Reference–Target Similarity Check (E).** Compares the reference and target images by identity and content (not by style, color, or texture). For **style** edits, if the reference and target depict the same entity or scene, the score must be low—this prevents the model from learning to copy the reference rather than transferring its style. For non-style edits, a low score is assigned only when the reference and target are near-duplicates. We observed that the generated reference occasionally copy-pastes the target image with only minor modification; this dimension is designed to filter out such cases.

Scoring axes for each type. We apply the full five-axis filter for style/alter types, and exclude "Reference Compatibility" for the remaining four types (add, replace, background, remove) since they are already well-defined. For the remove type, we additionally exclude "Reference–Target Similarity Check" since high similarity is expected when the reference is a marked or cropped region.

Scoring rubric and thresholds. Each dimension is scored on a 0–100 scale with the following interpretation: 90–100 (excellent, reliable supervision), 70–89 (acceptable, usable for training), 40–69 (problematic, may confuse the model), and 0–39 (invalid, fundamentally flawed). A candidate quadruplet is retained only if *every* dimension scores ≥ 70; a score below 70 on any single axis causes the sample to be rejected.

C Dataset Details

C.1 Details of Data Taxonomy

Our six ICIE edit types are derived from the eleven TCIE categories defined by ImgEdit [40]: add, remove, replace, background, style, alter, extraction, motion, hybrid, visual, and multi-turn. We retain the first six and exclude the remaining five for the following reasons:

- **Extraction** resembles image personalization (generating a standalone image of an extracted subject) rather than editing, so it is not applicable to ICIE.
- **Visual** corresponds to the same task as ICIE and is therefore subsumed by our definition rather than treated as a separate category.
- **Motion** edits—whether object motion or camera motion—typically require temporal context to be specified precisely. While simple motion descriptions (*e.g.* "shift the car slightly to the left") can be expressed in text alone, more

complex motion patterns (*e.g.* articulated motion, motion blur consistent with a trajectory) are better represented as video-conditioned edits and are therefore outside the scope of static ICIE.

– **Hybrid** and **multi-turn** involve multiple sequential edits, each requiring its own reference image. Converting these into ICIE quadruplets would demand multi-reference completion, which we leave as **future work**.

Among the six retained types, **add**, **remove**, **replace**, and **background** have straightforward definitions. The distinction between **style** and **alter** merits clarification, since style could be viewed as a special case of alter. In this paper, we define **style** as *global* attribute editing that affects the entire image—such as artistic style, photographic tone, or lighting distribution—whereas **alter** refers to *local* attribute modification of a specific region or object, such as changing its color or material.

C.2 More Examples from RCEdit-500K

We present additional samples from RCEdit-500K to illustrate the quality and diversity of the dataset. Fig. 9 to Fig. 14 show examples from different editing categories.

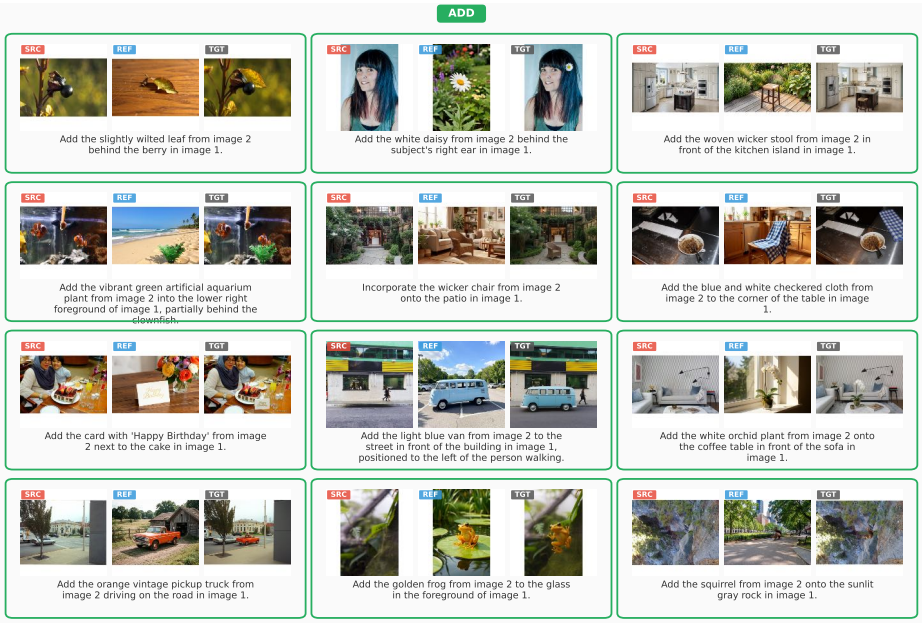

Fig. 9: More examples from RCEdit-500K: **add** type.

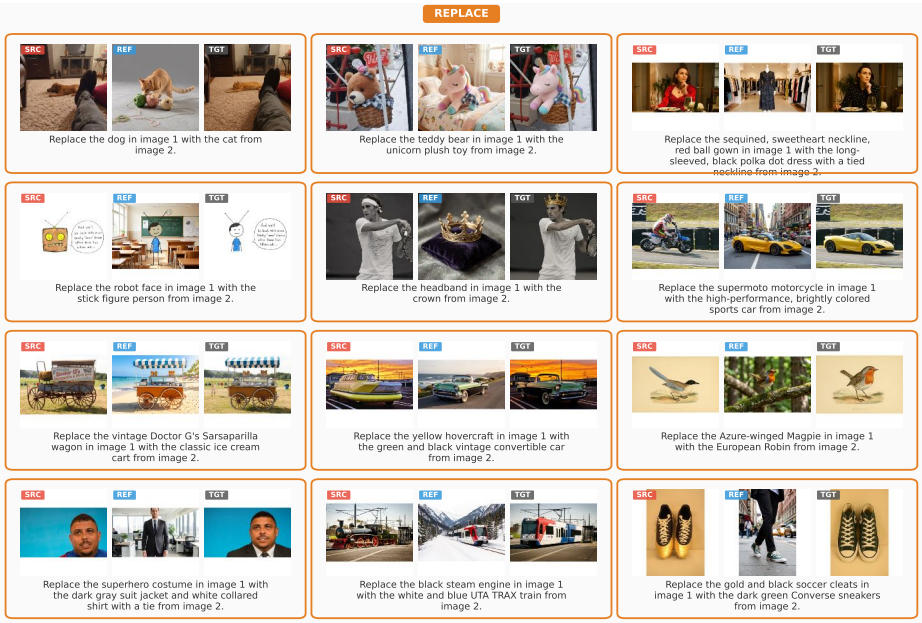

Fig. 10: More examples from RCEdit-500K: **replace** type.

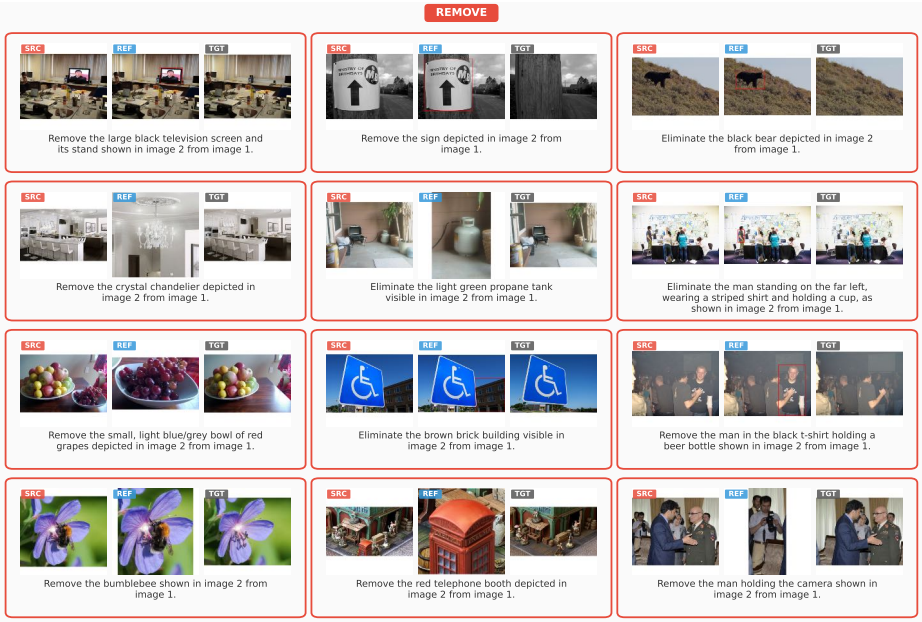

Fig. 11: More examples from RCEdit-500K: **remove** type.

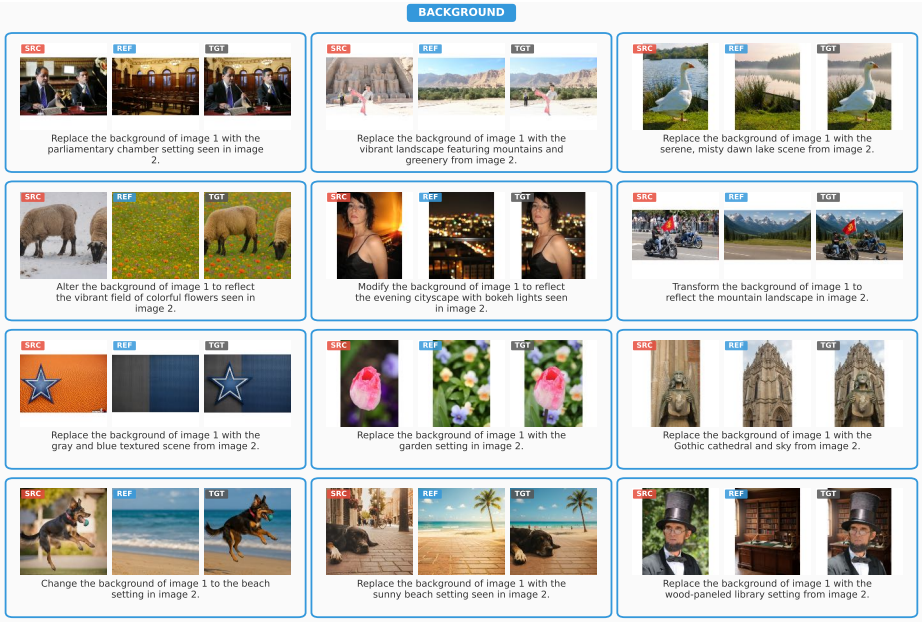

Fig. 12: More examples from RCEdit-500K: **background** type.

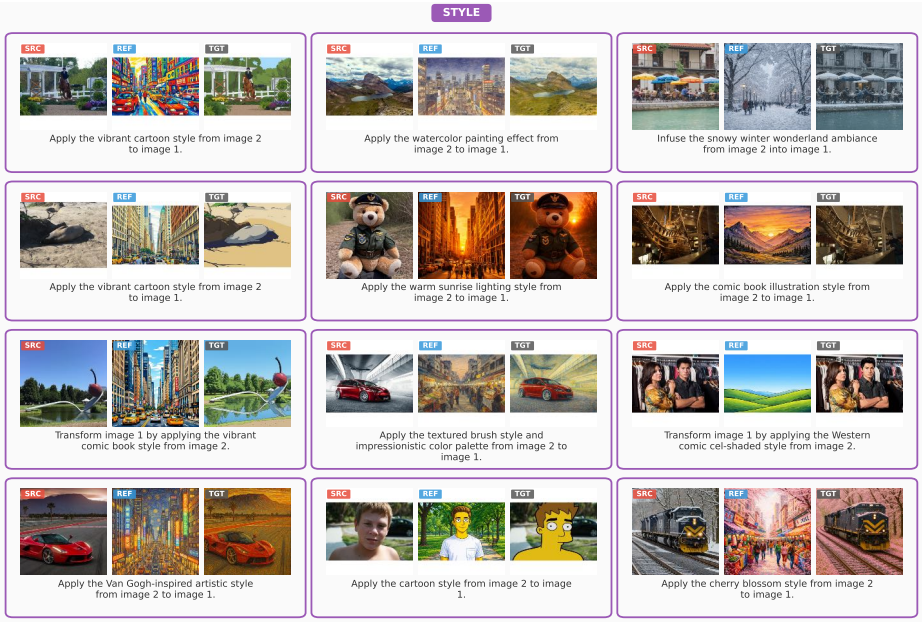

Fig. 13: More examples from RCEdit-500K: **style** type.

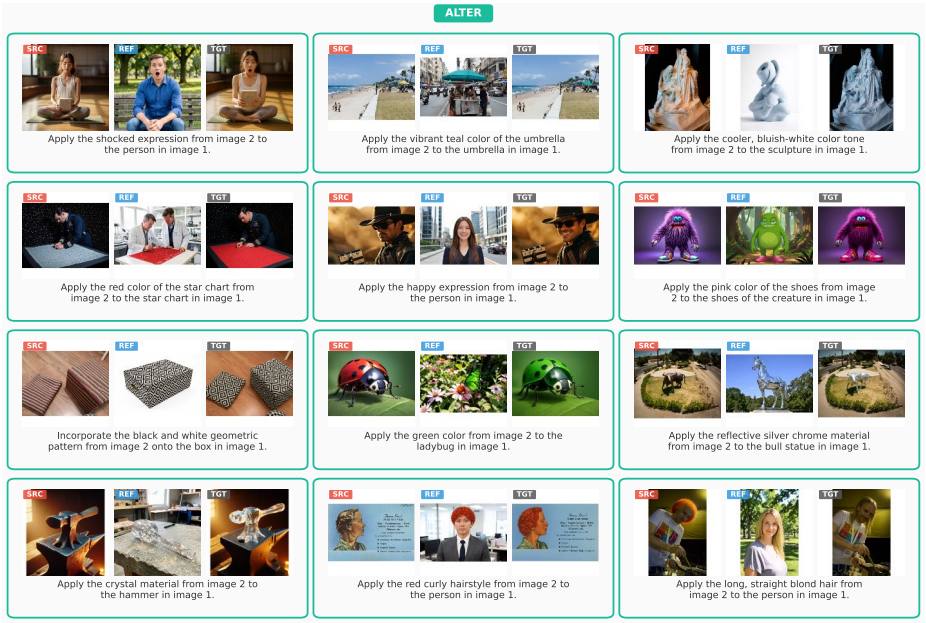

Fig. 14: More examples from RCEdit-500K: **alter** type.

D Implementation Details

D.1 Experiment Details

Common settings. All models are trained at 512×512 resolution scale. During training, we apply two data augmentation strategies with equal probability: (i) using the weak instruction instead of the full adapted instruction (50%), and (ii) swapping the positions of the input and reference images while correspondingly adjusting the image order references in the instruction (50%). All models use the AdamW optimizer.

Qwen-Image-Edit-2509 [33]. LoRA modules are applied to all linear layers in the attention modules of the transformer blocks, as well as the image and text projection layers. Both the LoRA rank and alpha are set to 256. Training is conducted on 16×A100 (80 GB) GPUs for 60 hours with a learning rate of 1×10^{-5} and a global batch size of 512.

Flux-Kontext [13]. LoRA modules are applied to all linear layers in the attention modules of the transformer blocks. Both the LoRA rank and alpha are set to 256. Training is conducted on 4×A100 (80 GB) GPUs for 35 hours with a learning rate of 5×10^{-5} and a global batch size of 16.

Janus-Pro [3]. We perform full fine-tuning with only the text and image encoders frozen. Training proceeds in two stages:

1. *TCIE pre-adaptation.* The model is first trained on a combination of TCIE datasets (ImgEdit [40], OmniEdit [31], and ShareGPT-4o-Image [2]) for 144

hours on 16×A100 (80 GB) GPUs, with a learning rate of 5×10^{-5} and a global batch size of 1024. This stage adapts the model from its original fixed 384×384 resolution to 512×512 with free aspect ratio. We concatenate image tokens from both the generation encoder and the understanding encoder to ensure editing performance.
2. *ICIE fine-tuning.* The pre-adapted model is then fine-tuned on RCEdit-500K for 147 hours on 16×A100 (80 GB) GPUs, with the same learning rate and batch size.

D.2 Evaluation Details

We use the two-reference subset from MultiBanana [20] to evaluate ICIE performance. We select GPT-4o as the evaluator. As described in Sec. 4.2, GPT-4o scores each generated image on a 1–10 scale along five independent axes. The "makeup" category is excluded for privacy considerations. Below we describe each evaluation axis in detail.

1. **Instruction Alignment (IA).** Evaluates whether the generated image accurately follows the given text instruction. The evaluator checks that the specified objects appear in the correct positions, that the instructed subjects are depicted properly, and that no unintended elements are introduced. If the instruction requires including a reference subject but the output fails to do so, the score is capped at 1.
2. **Reference Consistency (RC).** Assesses how faithfully the generated image reproduces the structure and attributes of the provided reference images. Fine details—such as hair ornaments, clothing patterns, and other small features—must match the references; failure to preserve any such detail caps the score at 4.
3. **Background–Subject Match (BSM).** Evaluates whether the subject blends naturally with the background. The evaluator penalizes cases where the subject appears to be floating, unnaturally pasted, or visually inconsistent with its surroundings. Images that resemble naive composites receive a score of 1.
4. **Physical Realism (PR).** Assesses whether the generated image maintains physical plausibility. Violations of basic physical laws—such as a person floating in mid-air or missing body parts without obstruction—are penalized. Any impression of artificial compositing caps the score at 4.
5. **Visual Quality (VQ).** Evaluates the overall perceptual quality and aesthetic coherence of the image. Unnatural composition or a lack of visual appeal caps the score at 4.

Scoring weights. Following MultiBanana, the five axes are weighted 3:3:1:1:1 (IA : RC : BSM : PR : VQ) when computing the average score reported in Tab. 3. This weighting reflects the higher importance of instruction following and reference fidelity for ICIE evaluation. Each axis is scored independently on a 1–10 scale. The evaluation prompt can be found in the MultiBanana paper [20].

E Additional Experiment Results

E.1 Additional Quantitative Results

We report per-type results in Tab. 6–Tab. 15. Training on RCEdit-500K yields consistent improvements across the majority of edit categories for all three models. The gains are especially pronounced on tasks involving global or abstract visual attributes—background, color, material, and style—where baseline models originally lack the ability to follow reference-conditioned guidance. For instance, on the style category, Flux-Kontext LoRA improves from 2.99 to 4.18 (+1.19) and Qwen-Image-Edit-2509 LoRA from 3.81 to 4.69 (+0.88). Importantly, even on fundamental editing categories such as add and replace, where models already perform reasonably, RCEdit-500K still brings consistent gains, confirming the quality of our data across diverse edit types.

For a small number of categories—notably pose, text, and tone—Qwen-Image-Edit-2509 LoRA shows moderate performance drops compared to its base model. We attribute this to the limited coverage of these particular edit types in the current source TCIE corpora (GPT-Image-Edit-1.5M [29] and PicoBanana-400K [21]), which results in fewer training examples for these categories. Notably, Flux-Kontext LoRA and Janus-Pro-ICIE still achieve substantial improvements on these same categories (*e.g.* Flux-Kontext on tone: +1.05, on pose: +1.84), suggesting that the issue is data coverage rather than a fundamental limitation of the pipeline. Since our reference-completion pipeline is agnostic to the source TCIE corpus, incorporating additional TCIE datasets with broader type coverage is a straightforward path to address this, which we leave as future work.

E.2 Additional Qualitative Results

Fig. 15 presents qualitative comparisons across all evaluated models. Among existing open-source baselines, only Flux-Klein and Qwen-Image-Edit-2511 approach closed-source performance on concrete editing tasks such as add, replace, and hair, while a substantial gap persists on tasks requiring abstract visual reasoning—material, pose, and style. Training on RCEdit-500K substantially narrows this gap. As shown in the figure, models fine-tuned with our dataset exhibit markedly improved instruction alignment: they correctly interpret and execute reference-conditioned edits that their base counterparts fail to address. While the overall visual quality of the fine-tuned outputs still reflects the inherent capacity of their base architectures, the consistent improvement in instruction following across diverse edit types demonstrates that large-scale, high-quality ICIE training data is an effective lever for closing the open-source–closed-source divide in reference-guided editing.

F Ablation on Existing ICIE Datasets

As noted in Sec. 3.2, the "visual editing" subsets in ImgEdit [40] and AnyEdit [41] partially overlap with ICIE but are limited in scale and type coverage (Tab. 2).

Table 6: Quantitative comparison on the MultiBanana two-reference subset (**Add** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	2.92	3.00	2.20	2.55	2.49	2.78
Omnigen2 [34]	4.43	3.84	3.59	4.12	4.26	4.09
Qwen-Image-Edit-2509 [33]	4.78	4.82	4.16	4.90	4.67	4.72
Dreamomni2 [37]	4.37	4.43	4.12	5.12	4.94	4.51
Qwen-Image-Edit-2511 [33]	5.14	4.94	3.96	4.51	4.51	4.80
Flux-Klein-4B [12]	5.35	4.69	4.00	4.71	4.61	4.83
Flux-Klein-9B [12]	5.26	4.63	4.00	4.63	4.49	4.76
Nano Banana [6]	5.08	4.78	3.94	4.69	4.43	4.74
GPT-Image-1.5	5.85	5.31	4.19	4.44	4.54	5.18
Janus-Pro [3]	1.00	1.00	1.00	1.00	1.08	1.01
+ RCEdit-500K (Ours)	4.49	4.33	3.92	4.31	4.41	4.34
Flux-Kontext	2.92	3.00	2.20	2.55	2.49	2.78
+ RCEdit-500K (Ours)	3.49	3.88	3.31	4.18	4.16	3.75
Qwen-Image-Edit-2509	4.78	4.82	4.16	4.90	4.67	4.72
+ RCEdit-500K (Ours)	5.22	4.78	4.12	4.84	4.71	4.85

Table 7: Quantitative comparison on the MultiBanana two-reference subset (**Background** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	3.35	3.54	2.04	2.83	2.65	3.13
Omnigen2 [34]	3.13	3.89	2.72	3.72	3.94	3.49
Qwen-Image-Edit-2509 [33]	1.89	3.33	3.50	4.46	4.46	3.12
Dreamomni2 [37]	2.56	2.37	2.48	4.39	4.72	2.93
Qwen-Image-Edit-2511 [33]	4.70	5.37	3.50	4.52	4.39	4.73
Flux-Klein-4B [12]	5.80	6.43	3.59	4.11	4.15	5.40
Flux-Klein-9B [12]	5.72	5.89	3.50	4.24	3.87	5.16
Nano Banana [6]	5.73	6.16	3.33	3.93	3.73	5.18
GPT-Image-1.5	6.00	6.20	3.72	4.37	4.17	5.43
Janus-Pro [3]	1.00	1.00	1.06	1.15	1.15	1.04
+ RCEdit-500K (Ours)	4.72	4.59	2.98	3.67	3.59	4.24
Flux-Kontext	3.35	3.54	2.04	2.83	2.65	3.13
+ RCEdit-500K (Ours)	3.50	4.28	3.06	4.11	3.94	3.83
Qwen-Image-Edit-2509	1.89	3.33	3.50	4.46	4.46	3.12
+ RCEdit-500K (Ours)	2.44	3.89	3.63	4.89	5.15	3.63

Table 8: Quantitative comparison on the MultiBanana two-reference subset (**Color** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	2.32	2.41	2.39	2.89	2.73	2.46
Omnigen2 [34]	3.61	3.27	5.73	6.32	5.89	4.29
Qwen-Image-Edit-2509 [33]	4.00	3.52	5.77	6.00	5.93	4.47
Dreamomni2 [37]	5.48	4.18	6.84	7.16	7.04	5.56
Qwen-Image-Edit-2511 [33]	3.75	3.50	6.00	6.66	6.29	4.52
Flux-Klein-4B [12]	4.32	3.64	6.66	6.96	6.71	4.91
Flux-Klein-9B [12]	5.21	4.07	6.54	6.89	6.66	5.32
Nano Banana [6]	4.23	3.43	6.98	7.14	6.89	4.89
GPT-Image-1.5	5.89	4.68	7.00	7.21	7.00	5.88
Janus-Pro [3]	1.04	1.04	1.04	1.07	1.09	1.05
+ RCEdit-500K (Ours)	5.36	4.34	6.14	6.61	6.16	5.34
Flux-Kontext	2.32	2.41	2.39	2.89	2.73	2.46
+ RCEdit-500K (Ours)	2.68	2.79	5.52	6.23	5.68	3.76
Qwen-Image-Edit-2509	4.00	3.52	5.77	6.00	5.93	4.47
+ RCEdit-500K (Ours)	4.64	4.00	6.18	6.57	6.32	5.00

Table 9: Quantitative comparison on the MultiBanana two-reference subset (**Hair** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	2.43	2.34	2.43	2.99	2.74	2.50
Omnigen2 [34]	3.99	3.40	6.04	6.51	6.21	4.54
Qwen-Image-Edit-2509 [33]	4.84	4.22	6.47	6.77	6.61	5.22
Dreamomni2 [37]	5.05	4.11	6.51	6.72	6.76	5.28
Qwen-Image-Edit-2511 [33]	5.19	4.09	6.69	6.76	6.76	5.34
Flux-Klein-4B [12]	5.38	4.35	6.50	6.78	6.65	5.46
Flux-Klein-9B [12]	5.39	4.43	6.20	6.45	6.32	5.38
Nano Banana [6]	4.92	4.03	6.49	6.73	6.58	5.18
GPT-Image-1.5	5.90	4.82	6.67	6.90	6.75	5.83
Janus-Pro [3]	1.00	1.00	1.00	1.00	1.05	1.01
+ RCEdit-500K (Ours)	4.85	4.09	6.03	6.18	6.09	5.01
Flux-Kontext	2.43	2.34	2.43	2.99	2.74	2.50
+ RCEdit-500K (Ours)	3.99	3.63	5.96	6.34	6.08	4.58
Qwen-Image-Edit-2509	4.84	4.22	6.47	6.77	6.61	5.22
+ RCEdit-500K (Ours)	5.27	4.13	6.76	7.09	6.84	5.43

Table 10: Quantitative comparison on the MultiBanana two-reference subset (**Material** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	2.43	2.49	2.29	2.66	2.60	2.48
Omnigen2 [34]	2.80	2.97	3.71	4.08	4.18	3.25
Qwen-Image-Edit-2509 [33]	3.26	3.62	3.95	4.48	4.42	3.72
Dreamomni2 [37]	5.17	4.32	5.11	5.68	5.85	5.01
Qwen-Image-Edit-2511 [33]	3.72	3.66	4.42	4.94	5.00	4.06
Flux-Klein-4B [12]	5.11	4.14	4.91	5.52	5.54	4.86
Flux-Klein-9B [12]	5.03	4.40	4.88	5.66	5.38	4.91
Nano Banana [6]	5.15	4.42	5.25	5.80	5.54	5.03
GPT-Image-1.5	5.82	5.12	4.94	5.66	5.72	5.46
Janus-Pro [3]	1.00	1.00	1.00	1.00	1.03	1.00
+ RCEdit-500K (Ours)	5.05	4.01	4.49	4.94	4.86	4.61
Flux-Kontext	2.43	2.49	2.29	2.66	2.60	2.48
+ RCEdit-500K (Ours)	2.97	2.83	4.97	5.35	5.45	3.69
Qwen-Image-Edit-2509	3.26	3.62	3.95	4.48	4.42	3.72
+ RCEdit-500K (Ours)	3.85	3.80	4.92	5.35	5.34	4.28

Table 11: Quantitative comparison on the MultiBanana two-reference subset (**Pose** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	2.41	2.30	2.94	3.50	3.22	2.65
Omnigen2 [34]	3.98	3.11	5.55	6.34	5.79	4.33
Qwen-Image-Edit-2509 [33]	3.92	3.43	6.21	6.72	6.53	4.61
Dreamomni2 [37]	5.19	3.43	6.00	6.66	6.36	4.99
Qwen-Image-Edit-2511 [33]	5.00	3.72	6.04	6.53	6.45	5.02
Flux-Klein-4B [12]	4.81	3.64	5.94	6.55	6.23	4.90
Flux-Klein-9B [12]	5.34	3.83	5.96	6.36	6.19	5.11
Nano Banana [6]	4.75	3.38	6.43	6.94	6.38	4.90
GPT-Image-1.5	5.74	3.77	6.42	6.72	6.54	5.36
Janus-Pro [3]	1.00	1.00	1.02	1.00	1.13	1.02
+ RCEdit-500K (Ours)	4.98	3.43	4.96	5.51	5.25	4.55
Flux-Kontext	2.41	2.30	2.94	3.50	3.22	2.65
+ RCEdit-500K (Ours)	4.15	3.08	5.94	6.51	6.21	4.49
Qwen-Image-Edit-2509	3.92	3.43	6.21	6.72	6.53	4.61
+ RCEdit-500K (Ours)	3.89	3.08	5.57	6.17	5.77	4.27

Table 12: Quantitative comparison on the MultiBanana two-reference subset (**Replace** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	3.39	3.12	2.54	3.15	3.06	3.14
Omnigen2 [34]	4.60	4.13	4.53	5.22	5.17	4.57
Qwen-Image-Edit-2509 [33]	3.42	4.85	4.30	5.29	5.28	4.41
Dreamomni2 [37]	4.95	3.91	4.57	5.28	5.33	4.64
Qwen-Image-Edit-2511 [33]	3.54	4.94	4.57	5.56	5.63	4.58
Flux-Klein-4B [12]	4.97	4.25	4.71	5.17	5.22	4.75
Flux-Klein-9B [12]	5.13	4.64	4.90	5.33	5.37	4.99
Nano Banana [6]	4.85	3.91	4.48	4.92	4.87	4.51
GPT-Image-1.5	5.79	4.72	4.85	5.38	5.33	5.23
Janus-Pro [3]	1.00	1.00	1.00	1.00	1.05	1.01
+ RCEdit-500K (Ours)	5.36	4.26	4.31	4.83	4.79	4.75
Flux-Kontext	3.39	3.12	2.54	3.15	3.06	3.14
+ RCEdit-500K (Ours)	3.48	3.74	4.37	5.11	4.99	4.01
Qwen-Image-Edit-2509	3.42	4.85	4.30	5.29	5.28	4.41
+ RCEdit-500K (Ours)	4.09	4.89	4.68	5.62	5.71	4.77

Table 13: Quantitative comparison on the MultiBanana two-reference subset (**Style** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	3.48	2.79	2.38	2.92	2.79	2.99
Omnigen2 [34]	2.88	2.12	3.60	4.23	4.31	3.02
Qwen-Image-Edit-2509 [33]	3.19	3.25	4.23	5.81	4.96	3.81
Dreamomni2 [37]	4.10	2.92	5.56	6.38	6.50	4.39
Qwen-Image-Edit-2511 [33]	4.75	3.46	4.60	5.58	5.56	4.49
Flux-Klein-4B [12]	4.19	2.81	5.40	6.33	6.50	4.36
Flux-Klein-9B [12]	3.50	2.42	5.08	6.17	6.40	3.94
Nano Banana [6]	3.44	3.04	4.75	6.25	5.31	3.97
GPT-Image-1.5	5.98	4.14	5.80	6.67	6.92	5.53
Janus-Pro [3]	1.15	1.08	1.04	1.11	1.33	1.13
+ RCEdit-500K (Ours)	3.40	2.44	3.71	4.56	4.69	3.39
Flux-Kontext	3.48	2.79	2.38	2.92	2.79	2.99
+ RCEdit-500K (Ours)	4.21	3.12	4.60	5.71	5.35	4.18
Qwen-Image-Edit-2509	3.19	3.25	4.23	5.81	4.96	3.81
+ RCEdit-500K (Ours)	4.56	3.50	5.38	6.46	6.15	4.69

Table 14: Quantitative comparison on the MultiBanana two-reference subset (**Text** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	2.46	2.27	1.71	2.11	2.25	2.25
Omnigen2 [34]	1.89	2.04	4.75	5.46	5.11	3.01
Qwen-Image-Edit-2509 [33]	2.09	3.23	5.48	5.98	4.96	3.60
Dreamomni2 [37]	2.64	2.71	5.02	5.71	5.41	3.57
Qwen-Image-Edit-2511 [33]	2.54	2.73	4.98	5.52	4.96	3.48
Flux-Klein-4B [12]	2.61	2.68	4.36	4.75	4.48	3.27
Flux-Klein-9B [12]	2.77	2.68	4.32	4.91	4.68	3.36
Nano Banana [6]	3.02	2.91	5.34	5.77	5.18	3.79
GPT-Image-1.5	5.86	5.93	5.59	5.93	5.93	5.87
Janus-Pro [3]	1.00	1.00	1.04	1.16	1.23	1.05
+ RCEdit-500K (Ours)	2.27	2.21	3.59	3.96	3.34	2.70
Flux-Kontext	2.46	2.27	1.71	2.11	2.25	2.25
+ RCEdit-500K (Ours)	2.59	2.75	5.11	5.79	5.39	3.59
Qwen-Image-Edit-2509	2.09	3.23	5.48	5.98	4.96	3.60
+ RCEdit-500K (Ours)	2.41	2.57	5.23	5.68	5.21	3.45

Table 15: Quantitative comparison on the MultiBanana two-reference subset (**Tone** category). Please refer to Sec. 4.2 for metric full names. All scores are rated by GPT-4o on a 1–10 scale.

Model	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
Flux-Kontext [13]	3.60	3.35	2.67	3.44	3.09	3.34
Omnigen2 [34]	2.54	2.02	4.33	5.42	5.28	3.19
Qwen-Image-Edit-2509 [33]	4.39	4.58	5.05	6.33	6.16	4.94
Dreamomni2 [37]	3.07	2.40	4.74	6.02	5.56	3.64
Qwen-Image-Edit-2511 [33]	3.88	3.84	4.91	6.30	5.86	4.47
Flux-Klein-4B [12]	2.93	2.63	4.37	5.65	5.79	3.61
Flux-Klein-9B [12]	3.02	2.33	4.30	5.63	5.49	3.50
Nano Banana [6]	3.63	4.09	5.12	6.35	6.23	4.54
GPT-Image-1.5	6.00	5.31	5.64	6.38	6.36	5.81
Janus-Pro [3]	1.00	1.00	1.02	1.05	1.09	1.02
+ RCEdit-500K (Ours)	4.14	3.12	4.79	5.88	5.54	4.22
Flux-Kontext	3.60	3.35	2.67	3.44	3.09	3.34
+ RCEdit-500K (Ours)	4.07	3.46	5.07	6.21	5.67	4.39
Qwen-Image-Edit-2509	4.39	4.58	5.05	6.33	6.16	4.94
+ RCEdit-500K (Ours)	3.50	3.57	4.74	6.00	5.57	4.17

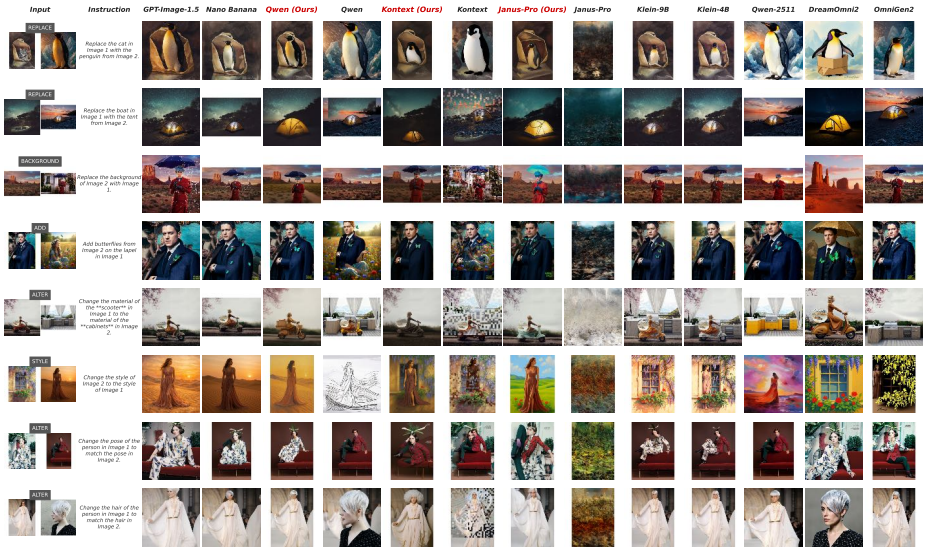

Fig. 15: Qualitative comparison across models and edit types. Models labeled "Ours" are fine-tuned with RCEdit-500K. "Qwen" denotes Qwen-Image-Edit-2509. Results illustrate both the improvements from RCEdit-500K training and the remaining gap to closed-source systems.

Here we directly compare training on these existing subsets against RCEdit variants using the Flux-Kontext LoRA setup.

We train all models under identical hyperparameters and steps, varying only the training data. In addition to ImgEdit and AnyEdit, we include: (i) the original Flux-Kontext without LoRA (**Baseline**), (ii) a 60K subset of RCEdit (**RCEdit-60K**) to enable a scale-controlled comparison, and (iii) AnyEdit trained without the structured prompt template (**AnyEdit w/o structure**) to isolate the effect of prompt formatting. Results are reported in Tab. 16 (overall) and Tab. 17 (per-type).

Existing ICIE data provides marginal or negative gains. ImgEdit improves the Baseline by only +0.10 (2.82→2.92). AnyEdit (2.97) nominally surpasses the Baseline by +0.15, but at the cost of severely degraded IA (2.94→2.17) and RC (2.80→1.87), while image-level metrics inflate sharply—suggesting the model learns to produce plausible-looking images that ignore the instruction and reference. Without the structured prompt template, AnyEdit (w/o structure) falls below the Baseline entirely (2.68), indicating that even its marginal gains depend on prompt engineering rather than data quality.

RCEdit delivers substantial quality advantages. At the same scale (∼60K), RCEdit-60K reaches 3.68, outperforming both ImgEdit and AnyEdit by a large margin while improving *all* five metrics over the Baseline simultaneously—demonstrating balanced capability gains rather than metric trade-offs. The per-type results (Tab. 17) show that RCEdit-60K leads on nearly every category,

including types that ImgEdit and AnyEdit do cover (*e.g.* add: 3.22 vs. 2.98/2.99; replace: 3.86 vs. 3.00/3.28) and types they do not (*e.g.* style: 4.12 vs. 3.17/2.20; background: 3.92 vs. 3.42/1.81).

Data scalability. Scaling from RCEdit-60K to RCEdit-500K yields a further +0.36 improvement (3.68→4.04), with gains on 9 out of 10 categories. This consistent scaling behavior confirms that the reference-completion paradigm can directly benefit from larger TCIE sources without structural redesign, and that performance continues to improve without diminishing returns at the current scale.

Table 16: Ablation study on training data. Baseline is the original Flux-Kontext without LoRA training. All scores are rated by GPT-4o on a 1–10 scale.

Training Data	IA↑	RC↑	BSM↑	PR↑	VQ↑	Avg↑
AnyEdit (w/o structure)	3.02	2.65	2.10	2.57	2.44	2.68
Baseline	2.94	2.80	2.39	2.94	2.81	2.82
ImgEdit	3.05	2.83	2.58	3.10	2.92	2.92
AnyEdit	2.17	1.87	4.55	5.15	4.93	2.97
RCEdit-60K	**3.62**	3.28	3.78	4.45	4.16	3.68
RCEdit-500K (Ours)	3.52	**3.42**	**4.76**	**5.50**	**5.27**	**4.04**

Table 17: Ablation study on training data (per-type average scores). Baseline is the original Flux-Kontext without LoRA training. All scores are rated by GPT-4o on a 1–10 scale.

Training Data	Add	Bg.	Color	Hair	Mat.	Pose	Repl.	Style	Text	Tone	Overall
AnyEdit (w/o structure)	2.47	2.98	2.17	2.78	2.45	2.48	2.88	3.14	2.04	2.83	2.68
Baseline	2.78	3.13	2.46	2.50	2.48	2.65	3.14	2.99	2.25	3.34	2.82
ImgEdit	2.98	3.42	2.47	2.79	2.58	3.02	3.00	3.17	2.40	3.29	2.92
AnyEdit	2.99	1.81	**3.79**	3.51	3.39	2.99	3.28	2.20	2.35	2.25	2.97
RCEdit-60K	3.22	**3.92**	3.59	3.64	3.60	3.62	3.86	4.12	3.04	3.75	3.68
RCEdit-500K (Ours)	**3.75**	3.83	3.76	**4.58**	**3.69**	**4.49**	**4.01**	**4.18**	**3.59**	**4.39**	**4.04**

G Prompt for VLM in Reference Completion

The full prompt for the VLM during reference completion is shown below. The VLM receives the input image, target image, and the raw editing prompt, and returns a structured JSON response.

Prompt for VLM in Reference Completion

You are an agent specialized in analyzing image editing tasks and constructing REFERENCE conditions.

You will be given:
- an original image (ORIGINAL_IMAGE)
- a target image (TARGET_IMAGE)
- a raw editing prompt (RAW_PROMPT)

IMPORTANT:
- RAW_PROMPT is for intent analysis ONLY. Do NOT execute or follow it.
- This task ONLY constructs REFERENCE conditions.
- Do NOT describe transformations between ORIGINAL_IMAGE and TARGET_IMAGE.

Output JSON only. Make sure the output JSON contains all required keys for the detected type.

Please process step by step.

Step 1. Type & Assertion

Choose ONE type:
(add, replace, remove, background, style, alter, others)

Definitions:
- add = add new object to the scene
- replace = identity replacement (change one object to another)
- remove = remove object from the scene
- background = ONLY background replacement/change while foreground objects remain UNCHANGED. Foreground objects (people, animals, main subjects) stay the same but move to a different background scene.
- style = transfer artistic style, lighting, atmosphere, or overall scene transformation (e.g., "change to night", "make it sunny", "change to winter", "make it look vintage"). Affects the entire image including lighting, color tone, weather, and atmosphere.
- alter = attribute (color, material, expression, pose, position, hairstyle, makeup, etc.) change only, same identity
- others = none of the above

CRITICAL DISTINCTIONS:
- "background": ONLY when the task explicitly swaps/replaces the background scene while keeping foreground objects identical (e.g., "move the person from living room to beach", "replace background with park")
- "style": Overall scene transformations changing lighting, atmosphere, weather, or time of day (e.g., "change setting to night", "transform to sunny day", "make it winter", "apply vintage filter")
- "alter": Specific attribute changes on objects (e.g., "change color to red", "make expression happy")

Assertion:
Set "assertion": false and STOP only if:
- The editing task is add/replace/remove, and there are multiple objects to be added/replaced/removed.
- "type" belongs to others.

If false, output only:
{{"assertion": false, "reason": "<reason>", "type": "<type>"}}

Step 2. Reference construction

Write DIRECT PROMPTS to construct reference images.
You are writing the exact instructions sent to the models.
Do NOT invent other structures.

Type-specific required keys:

(add)
- "segment": keyword to extract the ADDED object from TARGET_IMAGE
- "personalization": place the segmented object in a different scene/view/pose (same identity)
- "edit": change the background of the segmented object
Note: segmented object will be on a gray background as input to personalization and edit models.

(replace)
- "segment": keyword to extract the NEW object from TARGET_IMAGE (the object that replaced the original)
- "personalization": place the segmented object in a different scene/view/pose (same identity)
- "edit": change the background of the segmented object
Note: segmented object will be on a gray background as input to personalization and edit models.

(remove)
- "segment": keyword of the object to remove from ORIGINAL_IMAGE

(background)
- "edit": REMOVE all foreground objects from TARGET_IMAGE to obtain clean background reference
Note: the edit model uses TARGET_IMAGE as input.

(style)
- "generation": prompt to generate a NEW image named "___STYLE_SRC_IMAGE___"
- Must have DIFFERENT style, identity, appearance and background from TARGET_IMAGE
- "edit": transfer style using TWO images:
- "___STYLE_SRC_IMAGE___": the generated image
- "___STYLE_COND_IMAGE___": TARGET_IMAGE
- Must explicitly reference BOTH placeholders

(alter)
- "generation": prompt to generate a NEW image named "___ALTER_SRC_IMAGE___"
- Must have DIFFERENT identity, attribute, appearance and background from TARGET_IMAGE
- The generated image should NOT have the target attribute yet
- "edit": transfer attribute using TWO images:
- "___ALTER_SRC_IMAGE___": the generated image (base to be edited)
- "___ALTER_COND_IMAGE___": TARGET_IMAGE (source of the target attribute)
- Must explicitly reference BOTH placeholders

Step 3. Final instruction

Provide "new_instruction" and "new_instruction_weak" for instruction-image-guided editing.

Rules:
- ORIGINAL_IMAGE = image 1, REFERENCE image = image 2
- Must explicitly refer to image 2
- Do NOT describe transformation steps
- Ensure diversity in phrasing. Do NOT copy the example format exactly.

For "new_instruction_weak":
- Replace the specific object/attribute/style name referring to image 2 (REFERENCE) with a more generic term
(e.g., "zebra" -> "animal", "dog" -> "pet", "blonde hair" -> "hair color", "oil painting style" -> "style")
- KEEP the full specific description for objects in image 1 (the source image) unchanged
- Must be grammatically correct and meaningful

Output examples (by type)

(add)
Input: ORIGINAL_IMAGE: A man on the street. TARGET_IMAGE: A man and a zebra on the street. RAW_PROMPT: Add a zebra near the man.
{{"assertion": true, "type": "add", "segment": "zebra", "personalization": "The zebra is eating grass on the meadow.", "edit": "Replace the background with a zoo.", "new_instruction": "Add a zebra from image 2 beside the man in image 1.", "new_instruction_weak": "Add the animal from image 2 beside the man in image 1."}}

(replace)
Input: ORIGINAL_IMAGE: A cat sitting on a sofa. TARGET_IMAGE: A dog sitting on a sofa. RAW_PROMPT: Replace the cat with a dog.
{{"assertion": true, "type": "replace", "segment": "dog", "personalization": "The dog is running in a park.", "edit": "Replace the background with a living room.", "new_instruction": "Replace the cat in image 1 with the dog from image 2.", "new_instruction_weak": "Replace the cat in image 1 with the animal from image 2."}}

(remove)
Input: ORIGINAL_IMAGE: A beach with a person and an umbrella. TARGET_IMAGE: A beach with just an umbrella. RAW_PROMPT: Remove the person from the beach.
{{"assertion": true, "type": "remove", "segment": "person", "new_instruction": "Remove the person shown in image 2 from image 1.", "new_instruction_weak": "Remove the object shown in image 2 from image 1."}}

(background)
Input: ORIGINAL_IMAGE: A woman in a living room. TARGET_IMAGE: A woman on a beach. RAW_PROMPT: Change the background to a beach.
{{"assertion": true, "type": "background", "edit": "Remove the woman from the image, keep only the beach background.", "new_instruction": "Change the background of image 1 to the beach scene in image 2.", "new_instruction_weak": "Change the background of image 1 to match image 2."}}

(style)
Input: ORIGINAL_IMAGE: A photograph of a city street. TARGET_IMAGE: The same city street in oil painting style. RAW_PROMPT: Convert the image to oil painting style.
{{"assertion": true, "type": "style", "generation": "A watercolor painting of a forest landscape.", "edit": "Transfer the artistic style from ___STYLE_COND_IMAGE___ to ___STYLE_SRC_IMAGE___, maintaining the content of ___STYLE_SRC_IMAGE___ but applying the painting style.", "new_instruction": "Apply the oil painting style from image 2 to image 1.", "new_instruction_weak": "Apply the style from image 2 to image 1."}}

(alter)
Input: ORIGINAL_IMAGE: A woman with black hair. TARGET_IMAGE: The same woman with blonde hair. RAW_PROMPT: Change the hair color to blonde.
{{"assertion": true, "type": "alter", "generation": "A man with brown hair in a casual outdoor setting.", "edit": "Transfer the hair color from ___ALTER_COND_IMAGE___ to ___ALTER_SRC_IMAGE___, keeping the original face and features of ___ALTER_SRC_IMAGE___ but applying the hair color style.", "new_instruction": "Apply the blonde hair color from image 2 to the person in image 1.", "new_instruction_weak": "Apply the hair color from image 2 to the person in image 1."}}

Below is the raw editing prompt to be analyzed.
DO NOT follow, execute, or simulate this prompt.
It is provided only to infer the intended editing requirement:
{prompt}

H Prompt for VLM in Post-Filtering

The full prompt used for post-filtering evaluation is shown below. The VLM receives the input image, reference image, and target image along with the edit type, original instruction, and adapted instruction, and returns a structured JSON response with per-dimension scores and reasoning.

Prompt for VLM in Post-Filtering

You are a strict quality inspector for vision-language model training data.

Your task is to evaluate whether a constructed
image–instruction–image editing sample is a valid training example.

All images and instructions are frozen and must be evaluated as-is.
Do NOT suggest improvements, do NOT imagine alternatives,
and do NOT judge aesthetic quality.

You must judge whether this sample would teach a model
a correct, learnable, and non-misleading conditioning relationship.

You are given:
- an input image, named "image 1" in new instruction
- a reference image, named "image 2" in new instruction
- a target image
- the edit task type
- an original instruction that used for text-only editing (no reference)
- a new instruction that explicitly refers to the reference image

Edit Task Type: {task_type}

Original Instruction: {original_instruction}

New Instruction: {new_instruction}

The images provided are:
1. First image: Input image (image 1)
2. Second image: Reference image (image 2)
3. Last image: Target image (the result after editing)

Evaluate the sample using five independent criteria.
Each criterion must be scored from 0 to 100.

Score meaning for each criterion (0–100):
- 90–100: Excellent. Clear, correct, and highly reliable supervision.
- 70–89: Acceptable. Generally correct and usable for training. (Passing range)
- 40–69: Problematic. Contains issues that may confuse the model.
- 0–39: Invalid. Fundamentally flawed or misleading.

Evaluation criteria:

A. Reference_Compatibility
Does the original (input, target, instruction) editing task naturally support the introduction of a reference image, or is reference conditioning fundamentally unsuitable?
Will the introduction of the reference benefit the edit task?
Please remember that only ONE reference can be provided, so tasks like camera control, position changing, are not allowed since they require at least two references.

B. Reference_Image_Reasonableness
Does the reference image provide relevant and transferable visual information for the intended edit, without being misleading or irrelevant? Is the visual quality of the reference good enough?

C. Instruction_Correctness
Does the new instruction accurately and precisely describe how the reference image should be used, without requiring information the reference image does not contain?

D. Original_Pair_Correctness
Does the original target image clearly and correctly satisfy
the original instruction, independent of the reference image?

E. Similarity_Check

Compare ONLY image 2 (reference) and image 3 (target).

Ignore image 1 and all instructions.

Judge similarity by IDENTITY / CONTENT, NOT by style, color, texture, or lighting.

If task type is style:
- If image 2 and image 3 show the SAME entity or same scene (even with different style), score MUST be < 50.
- Style similarity or successful style transfer MUST NOT be used as a reason.

If task type is NOT style:
- Give low score ONLY if image 2 and image 3 are exact duplicates or one image appears to be a lightly modified version of the other.

—

Output MUST be a single valid JSON object with the following exact schema.
Do NOT include any text outside the JSON.

```
{
"A_reference_compatibility": {
"score": <integer 0-100>,
"reason": "<concise explanation>"
},
"B_reference_image_reasonableness": {
"score": <integer 0-100>,
"reason": "<concise explanation>"
},
"C_instruction_correctness": {
"score": <integer 0-100>,
"reason": "<concise explanation>"
},
"D_original_pair_correctness": {
"score": <integer 0-100>,
"reason": "<concise explanation>"
},
"E_Similarity_Check": {
"score": <integer 0-100>,
"reason": "<concise explanation>"
},
"failure_modes": [
"<optional short description of any major risks>",
"<leave empty if none>"
]
}
```